

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

1. 資料概要

1.1. 概要

本資料(以降アプリケーションノートと記載)は、ルネサス製USB汎用ASSPデバイス M66291を制御するための応用技術資料です。

1.2. アプリケーションノートの特徴

アプリケーションノートは以下のような特徴を持っています。

- ・M66291 制御に関する手法を、記述例をあげて具体的に記載しています。
- ・M66291 を使用する場合の注意事項を記載しています。
- ・「ルネサス汎用 ASSP M66291/M66290A USB Sample Firmware 」とリンクしています。
- ・豊富な機能用途を具体的に記載しています。

なお、「ルネサス汎用ASSP M66290A/M66291 USB Firmware」は一部のUSB特殊信号の制御には対応していません。また、本資料はM66291の制御方法の一例です。お客様の応用に合わせて変更する必要があります。

1.3. 関連ドキュメント

[1] M66291GP/HP データシート

【<http://www.renesas.com/jp/usb>】

[2] ルネサス汎用 ASSP M66291/M66290A USB Sample Firmware

【Renesas よりユーザに配布】

[3] USB Specification Revision 2.0

【<http://www.usb.org/developers/docs/>】

1.4. 凡例等

- (1) '*'を付加した端子名は"L"アクティブを示します。"L"アクティブ/"H"アクティブ設定可能な端子についても、デフォルトが"L"アクティブの場合は'*'を付加しています。
- (2) M66291 のレジスタ名称は、xxビット、xxレジスタ、のようにゴシックフォントを用いています。
- (3) 数値の記述方法は以下のとおりです。
十六進表記:00h、FFh など
十進表記:1、2、64、256 など
二進表記:'0000'、'0101'、'0'、'1'など

2. M66291 概要

2.1. M66291 の特徴

M66291 は、制御 CPU を特定しない USB ASSP です。

制御 CPU のメモリマップ上に M66291 のレジスタマップをアサインし、セパレートバスで直接 M66921 のレジスタへのアクセスを行う方法で制御します。接続方法については、[2.2 制御用 CPU との接続例](#)、及び、[7 応用例](#)を参照ください。

M66291 の主な特徴は以下のとおりです。詳細はデータシートを参照ください。

<p><<USB機能に関する特徴>></p> <ul style="list-style-type: none"> ◆ USB Specification Rev. 2.0準拠 ◆ USBトランシーバ回路内蔵 ◆ Full Speed(12Mbps)転送に対応 ◆ Vbus直結可能(5V耐圧入力)、D+信号プルアップ出力 ◆ オリジナルデザインの SIE(Serial Interface Engine)/DPLL (Digital Phase Lock Loop)による高品質USB信号 ◆ ハードウェアによるデバイスステート/コントロール転送の状態(遷移タイミング)管理により、エニュメレーションプログラム作成及びタイミング設計が容易 ◆ 連続送受信モード(複数のトランザクションデータをFIFOにバッファリング)により、CPU負荷軽減(ハイパフォーマンス、スループット向上) ◆ USB全転送タイプに対応(コントロール、バルク、アイソクロナス、インタラプト) ◆ エンドポイント用 FIFO バッファを 3KByte内蔵 new ◆ 最大7本のエンドポイント(EP0~6)を選択可能 ◆ エンドポイント毎(EP1~6)に以下の転送条件を設定可能 <ul style="list-style-type: none"> ● データ転送タイプ(バルク転送、アイソクロナス転送、インタラプト転送) ● 転送方向(IN/OUT) ● FIFOバッファサイズ(最大1024Byte×2) ● ダブルバッファ構成 ● 連続送受信(最大1024Byte×2のバッファリング) ● 最大パケットサイズ new ◆ SOF同期信号出力機能内蔵 	<p><<制御CPUとの接続性等の特徴>></p> <ul style="list-style-type: none"> new ◆ 4本のコンフィギュアブルFIFOポート <ul style="list-style-type: none"> ● エンドポイント番号の割り当て機能 new ◆ ビット幅(8ビット/16ビット)、エンディアン切替え new ◆ DMAインターフェースつき FIFO ポート2本 <ul style="list-style-type: none"> ◆ CPUへのUSB割り込み要因通知機能 ● 煩雑な要因解析を不要にする“割り込みキューイング機能” new ◆ 最大2本のINT端子 ◆ CPUとの柔軟な接続性 new ◆ バス幅(8ビット/16ビット)選択可能 new ◆ I/O電圧(2.7V~5.5V) new ◆ 割り込み信号(INT0、INT1)、DMA制御信号(Dreq0、Dreq1、Dack0、Dack1)の極性設定可能 new ◆ マルチワードDMA(バースト)可能 <ul style="list-style-type: none"> ◆ 4種のクロック入力周波数に対応 ● 6/12/24/48MHzクロック入力に対応 (IOVcc=2.7~3.6V時) ● 24/48MHzクロック入力に対応 (IOVcc=4.5~5.5V時) ● 発振バッファ、48MHz出力のPLL内蔵 new ◆ コア電源 3.3V単一、I/O電源 3~5V ◆ 低消費電力; <ul style="list-style-type: none"> ● 15mA(標準)(USB転送時) ● 10µA(標準)(クロック停止、USB切断時) new ◆ 15mA(標準)(USB転送時) new ◆ 10µA(標準)(クロック停止、USB切断時) <p><<用途>></p> <ul style="list-style-type: none"> ◆ プリンター、スキャナー ◆ ディスプレイモニタ、マルチメディアスピーカー ◆ デジタルカメラ、ターミナルアダプタ ◆ その他、Full Speed USB搭載のPC周辺機器全般 <p><<パッケージ>></p> <p>M66291GP :48P6Q (LQFP) M66291HP :52PJV(VQFN)</p>
--	--

new :M66290A からの追加/変更機能

図 2-1 M66291 の特徴

2.2. 制御用 CPU との接続例

図 2-2・図 2-4に M66291 を制御するための CPU としてルネサス 16 ビットマイコン M16C/80 を用いた接続例を示します。

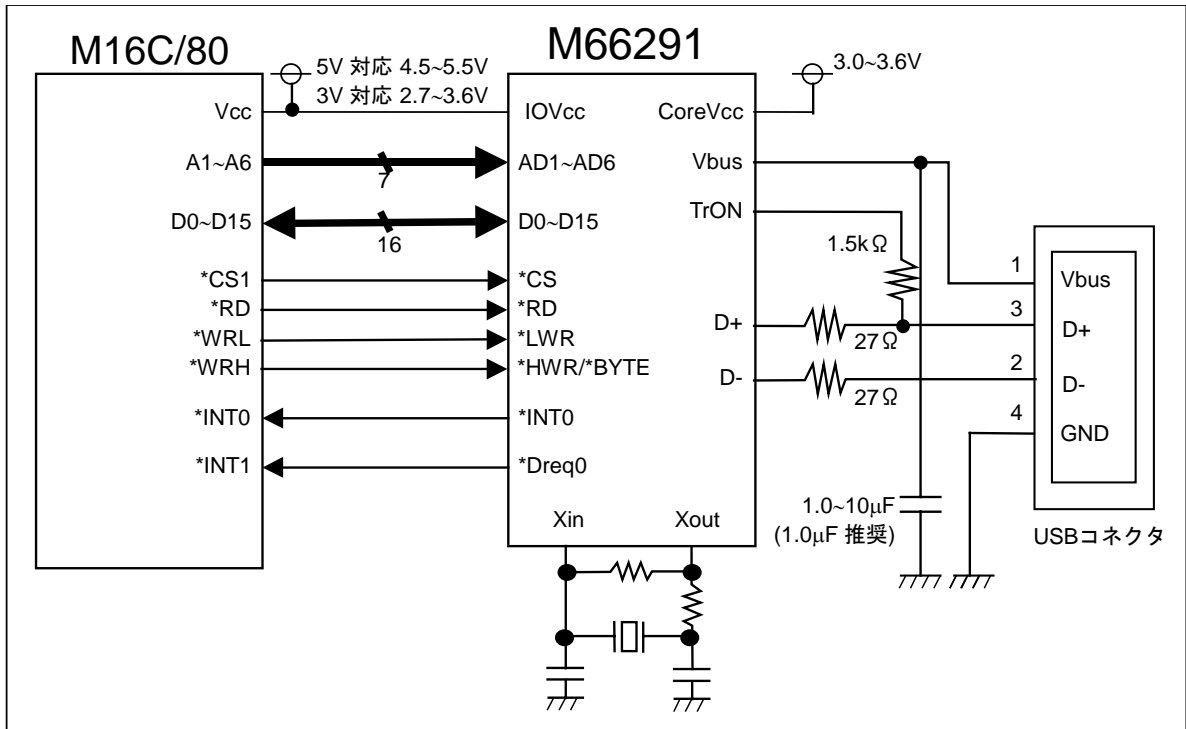


図 2-2 M66291 接続例 1 (16bit バス幅)

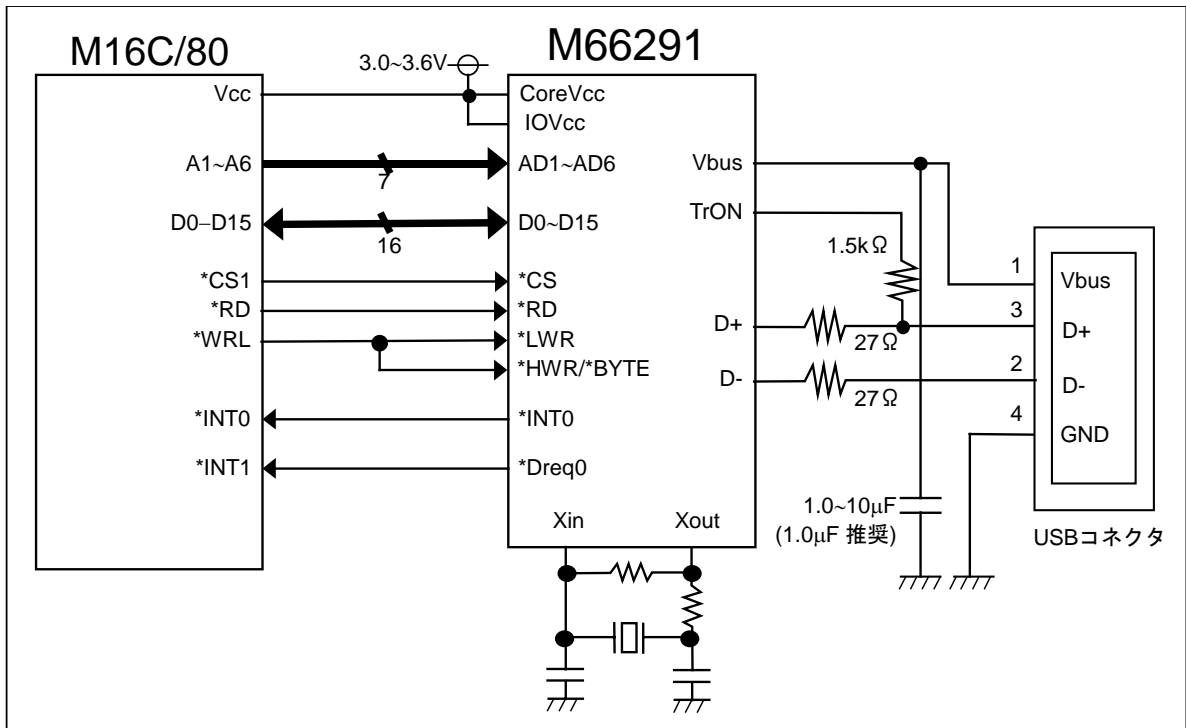


図 2-3 M66291 接続例 2 (16bit バス幅;M66290A 互換)

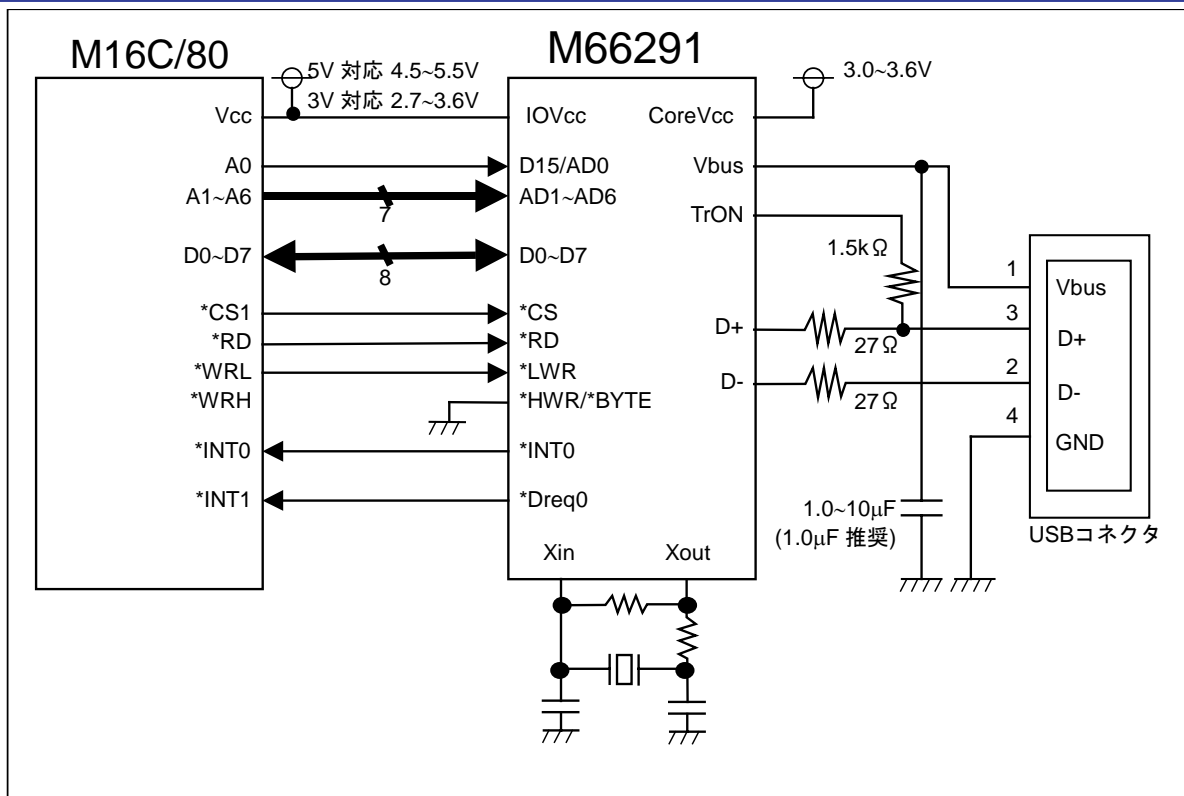


図 2-4 M66291 接続例 3 (8bit バス幅)

2.3. 制御用プログラム概略フロー

M66291 は USB のデータ送受信、特殊信号処理等を割り込み信号で CPU に通知します。したがって、制御プログラムは初期設定プログラムと USB 割り込み処理プログラムによって構成されます。

制御 CPU の外部割り込みプログラムにて、割り込み要因を判別し、該当する処理を実行してください。

(1) 特殊信号処理

Vbus 割り込み、レジューム割り込み、SOF 検出割り込み、デバイスステート遷移割り込み

(2) コントロール転送処理

コントロール転送ステージ遷移割り込み、デバイスステート遷移割り込み

(3) エンドポイントデータ転送処理

バッファエンpty/サイズオーバーエラー割り込み、バッファノットレディ割り込み、バッファレディ割り込み

M66291 が判定可能な各種 USB 処理を CPU の割り込みルーチンで処理する場合、M66291 制御プログラム構成例は以下のようになります。

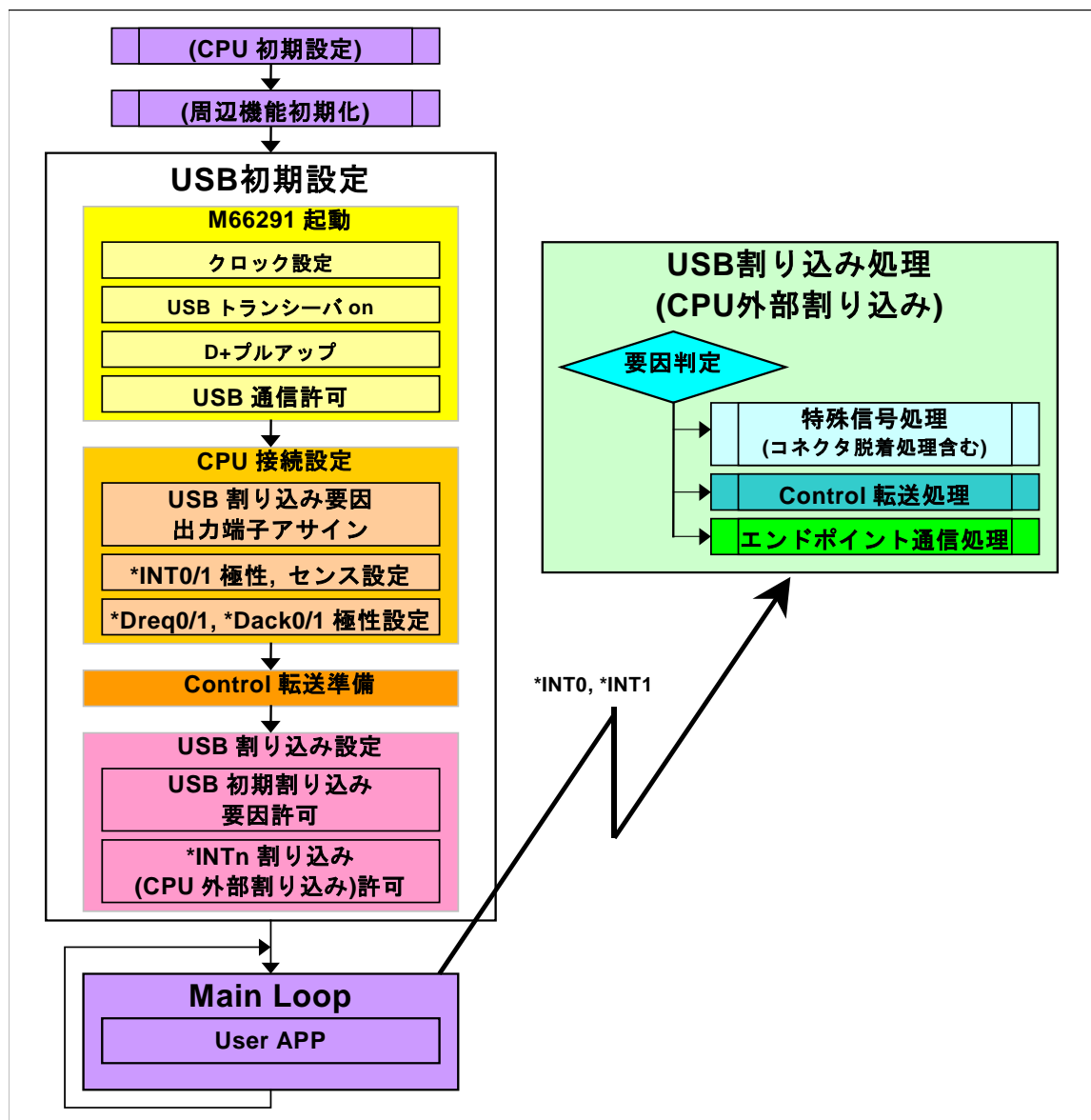


図 2-5 M66291 制御プログラム構成例

2.4. 割り込み処理

メイン CPU は M66291 からの *INTn 信号により USB 割り込みを認識します。USB 割り込みの要因は次の 4 種類の割り込みステータスレジスタの内容により識別可能です。

- (1) 割り込みステータス 0 レジスタ (Address 18h) 割り込み種別識別
- (2) 割り込みステータス 1 レジスタ (Address 1Ah) バッファレディ発生エンドポイント識別
- (3) 割り込みステータス 2 レジスタ (Address 1Ch) バッファノットレディ発生エンドポイント識別
- (4) 割り込みステータス 3 レジスタ (Address 1Eh) バッファエンプティ/サイズオーバーエラー発生エンドポイント識別

M66291 の割り込み識別には、割り込みステータス 0 レジスタ及び割り込み許可 0 レジスタを用います。各割り込みに対応したビット状態(二つのレジスタの論理積)により割り込み要因を認識し、該当する割り込み処理を行います。

割り込みステータス 1 レジスタ、割り込みステータス 2 レジスタ及び割り込みステータス 3 レジスタはエンドポイントバッファ状態割り込みに対応するエンドポイント番号を識別します。

必要に応じて、割り込み処理の中で各割り込み要因をクリアしてください。

USB 割り込みを検出した場合は下記のようなフローで処理を行います。

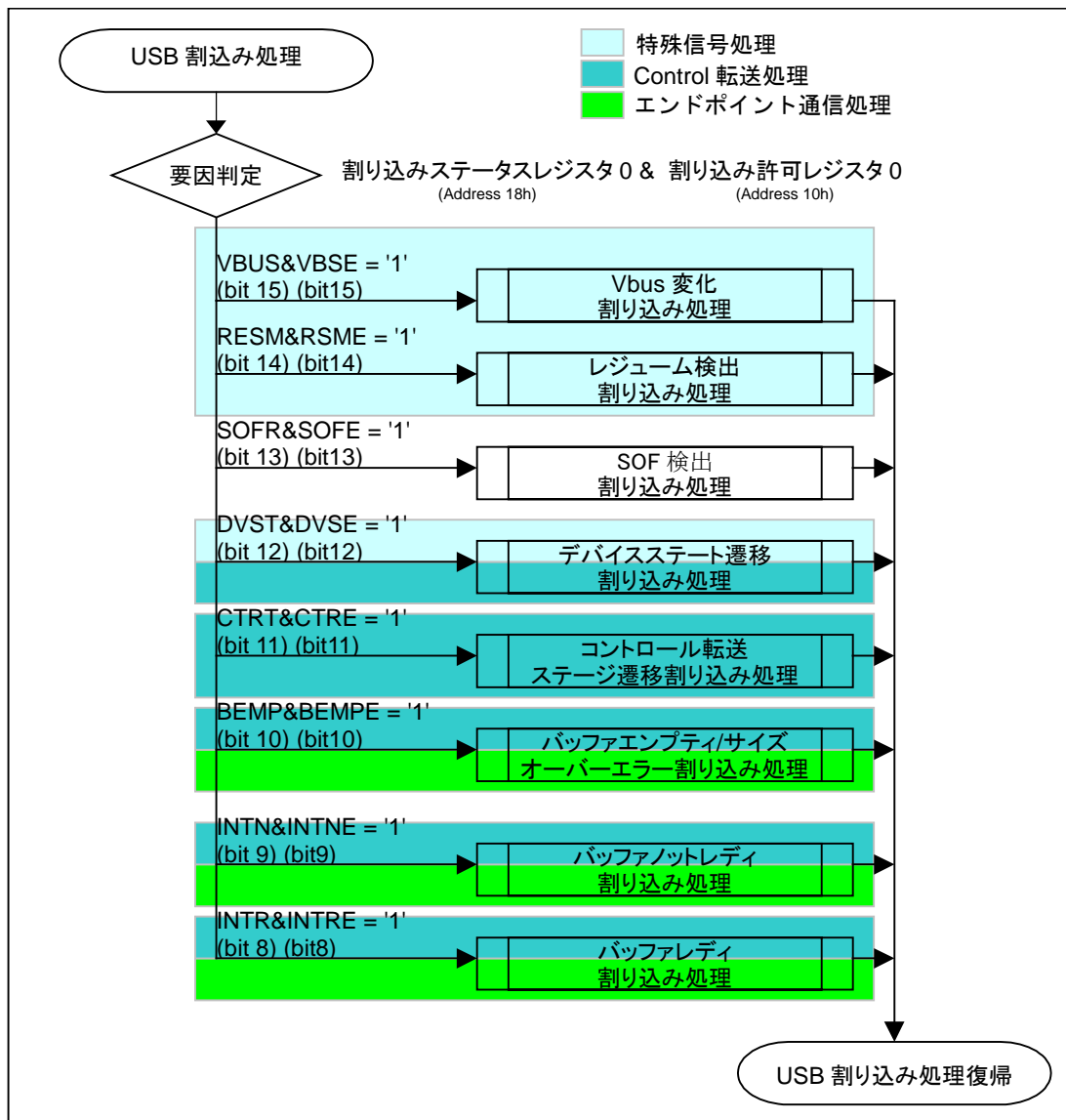


図 2-6 M66291 割り込み処理例

2.5. データ転送割り込み

M66291 は 3 タイプのデータ転送割り込み要因を備えており、ユーザの用途に合わせてデータ転送割り込み要因を選択することが可能です。各割り込み要因は割り込みステータスレジスタ 1~3 により、当該割り込み要因が発生したエンドポイントを認識することが可能です。各データ転送割り込み要因は転送タイプ、転送方向及びアクセス方法により割り込み発生条件が異なりますのでご注意ください。

各割り込みの要因クリア方法及び詳細機能については、「M66291 データシート」を参照ください。

表 2-1 データ転送タイプごとのデータ転送割り込み発生条件

転送タイプ		割り込み要因 (割り込み発生条件)	バッファレディ 割り込み (FIFO バッファ 読み書き可能)	バッファノット レディ割り込み (USB 送受信 不可)	バッファEMPTY/サイズ オーバーエラー割り込み (IN トークンに対し送信データ なし/最大パケットサイズを超 えるデータパケット受信)
コントロール Read 転送時			(発生しない)	IN トークンに対 し NAK 応答 ¹	送信完了
コントロール Write 転送時			受信データ読み 出し可能	OUT トークンに 対し NAK 応答	パケットサイズオーバ
CPU_FIFO_DATA へのアクセス	データ転送 IN 時	送信データ書き 込み可能	IN トークンに対 し NAK 応答 ¹	送信完了 (バッファEMPTY)	
	データ転送 OUT 時	受信データ読み 出し可能	OUT トークンに 対し NAK 応答 ¹	パケットサイズオーバ	
Dn_FIFO_DATA へのアクセス	データ転送 IN 時	送信データ書き 込み可能 ²	IN トークンに対 し NAK 応答	送信完了 (バッファEMPTY)	
	データ転送 OUT 時	受信データ読み 出し可能 ²	OUT トークンに 対し NAK 応答	パケットサイズオーバ	

¹ エンドポイントが IN 方向の場合に FIFO バッファをクリアを行うと、INTR 割り込みが連続して発生しますのでご注意ください。

² INTM[DMA 割り込みモードビット](Address 48h, 50h)でバッファレディ割り込みの発生有無を設定できます。あわせて、2.6 FIFO 制御も参照ください。

USB の各データ転送タイプごとの推奨設定は次のとおりです。

表 2-2 転送タイプ別推奨設定

転送タイプ	許可する割り込み要因	ダブルバッファ機能	連続転送機能
コントロール Read	コントロール転送ステージ遷移 バッファEMPTY/サイズオーバ エラー(非連続転送)	NA	O
コントロール Write	コントロール転送ステージ遷移 バッファレディ(非連続転送)	NA	O
バルク IN	バッファレディ	O	O
バルク OUT	バッファレディ	O	O
インタラプト IN	バッファEMPTY/サイズオーバ エラー	-	x
インタラプト OUT	バッファレディ	-	x
アイソクロナス IN	バッファレディ	O	x
アイソクロナス OUT	バッファレディ	O	x

O: 「使用する」設定を推奨
 - : 「使用しない」設定を推奨
 x : 必ず「使用しない」に設定
 NA: 「使用する」設定不可

2.6. FIFO 制御

M66291 は 3K バイトの大容量 FIFO バッファを内蔵しており、エンドポイント 1~エンドポイント 6 の各エンドポイントに対して別々のバッファ領域を割り当てて使用します。FIFO バッファは 64 バイト×48 ブロックで構成されており、各エンドポイントにおける領域確保は、EPI_Buf_Nmb[バッファ先頭番号ビット]¹、EPI_Buf_siz [バッファサイズビット]¹及び EPI_DBLB[ダブルバッファモードビット]¹で指定します。領域確保の方法は、後述の設定例を参照ください。領域は重複設定してもエラー扱いされません。重複領域に後から書き込まれたデータが有効となりますのでご注意ください。

各エンドポイントバッファへのアクセスは、4 本の FIFO データレジスタによって行います。FIFO データレジスタは、エンドポイント 0 専用の EP0_FIFO データレジスタ(CPU アクセス)、及び、エンドポイント 1~エンドポイント 6 共用の 3 本の FIFO データレジスタ構成され、共用の FIFO データレジスタは、CPU アクセス専用の CPU_FIFO データレジスタが 1 本、DMA 転送インターフェース可能な Dn_FIFO データレジスタが 2 本あります。EP1 - EP6 のどのエンドポイントへアクセスするかは、それぞれの FIFO 選択レジスタ²の設定に従います。

なお、CPU_EP[CPU アクセスエンドポイント指定ビット]²、D0_DMA_EP、D1_DMA_EP[DMA アクセスエンドポイント指定ビット]²で指定しているエンドポイント番号を重複させないでください。各 FIFO_DATA レジスタを使用しない場合は、CPU_EP または Dn_DMA_EP に"0h"を設定してください。

4 本の FIFO データレジスタはそれぞれ下記機能を内蔵しており、FIFO 選択レジスタ/制御レジスタの設定に従い動作を行います。

- (1) ショートパケット送信機能 (IVAL[IN バッファセット/OUT バッファステータスビット]³使用)
ショートパケット送出時等 FIFO バッファがフルになる前に送信データ書き込み完了としたい場合は、FIFO バッファにデータを書き込んだ後 IVAL='1'を書き込んでください。
- (2) 送信/受信バッファクリア機能 (BCLR[バッファクリアビット]³使用)
ダブルバッファ設定時は制御用 CPU からアクセス可能な面(片面)がクリアされます。USB バス側をクリアする場合は SCLR、もしくは、EPI_ACLR を使用してください。
- (3) zero-length パケット(Null パケット)送信機能 (IVAL & BCLR 使用)
IVAL='1'、BCLR='1'を同時に書き込むことで、zero-length パケット(Null パケット)送出の準備完了となります。ホスト PC からの IN トークンに対応して、M66291 は長さ zero-length パケット(Null パケット)を送信します。
- (4) データ長(8/16 ビット)設定機能 (Octl[レジスタ 8 ビットモードビット]² or EPI_Octl⁴使用)
16bit バス幅接続の場合に、FIFO バッファの書き込み途中で FIFO バッファレジスタの有効ビット数の切替え(16 ビット→8 ビット)が可能です。奇数バイト数((2n+1)バイト)データの書き込みを行いたい場合、16 ビットモード(Octl='0'、かつ、EPI_Octl='0')で 2n バイトのデータを書き込み、その後、Octl='1'(または、EPI_Octl='1')を設定することにより 8 ビットモードに切り替え、最後の 1 バイトのデータ書き込みを行います。⁵
- (5) 受信データ長カウントダウン機能 (RCNT[リードカウントモードビット]²使用)
RCNT='1'を設定して FIFO バッファの読み出しを行うと、1 回の読み出し毎に CPU_DTLN³がディクリメントされます。16 ビットアクセス時は-2 ずつ、8 ビットアクセス時は-1 ずつディクリメントされます。受信データ長カウントダウン機能は CPU_FIFO データレジスタへのアクセスの場合のみ有効です。
- (6) バッファリワインド機能 (RWND[バッファリワインドビット]²使用)
FIFO バッファにデータ書き込み途中、もしくは FIFO バッファからデータ読み出し途中に、RWND='1'を書きこむと、FIFO バッファ内部カウンタが初期化されます。
書き込み時:それまでに書き込みを行ったデータが無効となります。
読み出し時:バッファの先頭データから再読み出し可能となります。
RWND='0'であれば、書き込み/読み出し途中で CPU_EP/DMA_EP の切り替えを行い、再び CPU_EP を元のエンドポイントに割り付けた場合でも、続きのデータの書き込み/読み出しを行うことができ

¹ EPI コンフィギュレーションレジスタ 0(Address 60h、64h、68h、6Ch、70h、74h)

² CPU_FIFO 選択レジスタ 0(Address 40h)、Dn_FIFO 選択レジスタ 0(Address 48h、50h)

³ CPU_FIFO 制御レジスタ 0(Address 42h)、Dn_FIFO 制御レジスタ 0(Address 4Ah、52h)

⁴ EPI コンフィギュレーションレジスタ 1(Address 62h、66h、6Ah、6Eh、72h、76h)

⁵ CPU_FIFO 選択レジスタ上の Octl と、CPU_EP で指定しているエンドポイントの EPI_Octl との"or"が動作に反映されます。

ます。¹

また、IN 方向のエンドポイントに EPI_NULMD⁴=1 を設定している場合、FIFO データ書き込み中は“バッファノットレディ状態(ホストからの IN トークンに応答できないため)”と認識され、zero-length パケット(Null パケット)送信自動付加機能が動作します。

FIFO バッファ書き込みが遅い場合は上記のような現象が発生しますのでご注意ください。なお、M66291 は、以下の(1)または(2)のときに FIFO バッファ書き込み完了と認識します。

- (1) IVAL=1 を書き込んだ。((書き込みデータ数)<(FIFO バッファサイズ)の場合)
- (2) FIFO バッファ Full まで書き込みを行った。

連続転送設定時、(送信したいデータ数)<(FIFO バッファサイズ)の場合にも、上記(1)の方法で書きこみ終了させてください。

FIFO バッファのアクセスフローの例は [5 データ転送](#) を参照ください。

2.6.1. 連続転送モード時に、FIFO バッファ容量未満かつ最大パケットサイズの整数倍のデータを読み出す方法

連続転送モード時に、CPU_FIFO データレジスタ経由で、FIFO バッファ容量未満かつ最大パケットサイズの整数倍のデータをリードする場合の手順例を示します。

USB specification revision1.1 の 5.8.3 章で、bulk 転送のデータ転送完了は次のように定義されており、ここでは、前者の場合(デバイス側が受信データ数を予測可能な場合)について例を示します。

- Has transferred exactly the amount of data expected
- Transfers a packet with a payload size less than *wMaxPacketSize* or transfers a zero-length packet.

この場合には、バッファレディ割り込みでデータを読み出すたびに受信すべき総受信バイト数と、受信済みの総バイト数を比較し、最後の 1 面受信時には SIE_DTLN[SIE_FIFO 受信データ長ビット]² をポーリングすることにより、受信完了を知ることができます。SIE_DTLN は 1 パケット分のデータを受信完了したタイミングでハードウェアにより更新されます。SIE_DTLN が受信すべきバイト数に達した時点で、以下の手順でデータを読み出してください。

- (1) CPU_EP で指定したエンドポイントの PID³ = '00'(NAK)にする
- (2) Sreq²=0 を確認する(Sreq=0 になるまで待つ)
- (3) TGL²=1 を書き込む(受信データが CPU から読み出し可能になる)
- (4) Creq⁴=0 を確認する(Creq=0 になるまで待つ)
- (5) データ読み出し
- (6) 読み出し完了後、CPU_EP で指定したエンドポイントの PID = '01'(Buf)にする

¹ RWND=1 書き込み時は、RWND と CPU_FIFO 選択レジスタ/Dn_FIFO 選択レジスタ上の他のビットを同時に書き込まないでください。

² SIE_FIFO ステータスレジスタ 0(Address 46h)

³ EPI コンフィギュレーションレジスタ 1(Address 62h、66h、6Ah、6Eh、72h、76h)

⁴ CPU_FIFO 制御レジスタ 0(Address 42h)

表 2-3 各エンドポイントコンフィギュレーションレジスタ設定例 1

		転送タイプ			
		バルク IN	インタラプト IN	アイソクロナス IN	
使用例	最大パケットサイズ	64 バイト	32 バイト	100 バイト	
	FIFO バッファサイズ	512 バイト	64 バイト	256 バイト	
	ダブルバッファ構成	256×2	–	128×2	
	FIFO バッファ番号割り付け	08h–0Fh	20h–20h	22h–25h	
コンフィギュレーションレジスタ設定	0	転送タイプ [EPi_TYP]	'01'	'10'	'11'
		転送方向 [EPi_DIR]	'1'	'1'	'1'
		トグルモード [EPi_ITMD]	'0'	'0'	'0'
		バッファサイズ ¹ [EPi_Buf_siz]	'0011'(256 バイト)	'0000'(64 バイト)	'0001'(128 バイト)
		ダブルバッファモード [EPi_DBLB]	1b(256×2 バイト)	0b	1b(128×2 バイト)
		連続送受信モード [EPi_RWMD]	'1'	'0'	'0'
		バッファ先頭番号 [EPi_Buf_Nmb]	08h	20h	22h
	1	応答 PID [EPi_PID]	–	–	–
		長さ 0 のデータ送信自動付加モード ² [EPi_NULMD]	'1'	'0'	'0'
		自動クリアモード [EPi_ACLR]	'0'	'0'	'0'
		FIFO アクセスモード [EPi_Octl]	'0'	'0'	'0'
		最大パケットサイズ [EPi_MXPS]	040h	020h	064h

表 2-4 各エンドポイントコンフィギュレーションレジスタ設定例 2

		転送タイプ			
		バルク OUT	インタラプト OUT	アイソクロナス OUT	
使用例	最大パケットサイズ	64 バイト	32 バイト	100 バイト	
	FIFO バッファサイズ	1024 バイト	64 バイト	256 バイト	
	ダブルバッファ構成	512×2	–	128×2	
	FIFO バッファ番号割り付け	10h–1Fh	21h–21h	26h–29h	
コンフィギュレーションレジスタ設定	0	転送タイプ [EPi_TYP]	'01'	'10'	'11'
		転送方向 [EPi_DIR]	'0'	'0'	'0'
		トグルモード [EPi_ITMD]	'0'	'0'	'0'
		バッファサイズ ¹ [EPi_Buf_siz]	'0111'(1024 バイト)	'0000'(64 バイト)	'0001'(128 バイト)
		ダブルバッファモード [EPi_DBLB]	'1'(1024×2 バイト)	'0'	'1'(128×2 バイト)
		連続送受信モード [EPi_RWMD]	'1'	'0'	'0'
		バッファ先頭番号 [EPi_Buf_Nmb]	10h	21h	26h
	1	応答 PID [EPi_PID]	–	–	–
		長さ 0 のデータ送信自動付加モード ² [EPi_NULMD]	'0'	'0'	'0'
		自動クリアモード [EPi_ACLR]	'0'	'0'	'0'
		FIFO アクセスモード [EPi_Octl]	'0'	'0'	'0'
		最大パケットサイズ [EPi_MXPS]	040h	020h	064h

¹ 確保される FIFO バッファの領域=(EPi_Buf_siz+1)×64 バイトとなります。

² zero-length パケット(Null データ)送信自動付加モードを使用する場合の注意点

ダブル FIFO バッファの設定の場合に、バッファの片方のデータを PC に送信し終わった時点でもう片方にまだ制御 CPU からデータ書き込み中であつたら、M66291 は zero-length パケット(Null パケット)を自動的に送信してしまいます。よって、zero-length パケット(Null データ)送信自動付加モードは全部の送信データを FIFO バッファに書き込み終わったあとにセットする必要があります。

[9.6 zero-length パケット\(Null パケット\)送信自動付加機能に関する Q & A](#)も参照ください

2.7. Dn_FIFO データレジスタへのアクセス(DMA コントローラインターフェース)

2.7.1. DMA コントローラインターフェース概要

外部 DMA コントローラから FIFO バッファアクセスを行うために、M66291 は D0_FIFO データレジスタ/D1_FIFO データレジスタ(以下 Dn_FIFO データレジスタ)を持ちます。DMA コントローラによるアクセスを行う場合は、CPU アクセス同様に EPI コンフィギュレーションレジスタ 0/1 にてバッファ領域を確保し、Dn_FIFO 選択レジスタにて DMA 転送方法の指定を行い、Dn_FIFO データレジスタを使用してデータ転送を行います。

DMA 転送方法は、以下 4 つのパラメータによって設定します。ご使用になる DMA コントローラの仕様にあわせてパラメータを設定してください。パラメータ一覧を表 2-5 に、パラメータの組み合わせ一覧を表 2-6 に示します。各パラメータの詳細については、M66291GP/HP データシート【2.31 Dn_FIFO 選択レジスタ】を参照ください。

M66291 では、DMA コントローラを使用せずに(通常の CPU アクセスによって)、Dn_FIFO データレジスタに対してアクセスを行うことも可能です。

表 2-5 DMA インターフェース用パラメータ一覧

パラメータ	説明
バーストモード [BUST]	サイクルスチール転送/バースト転送の選択 '0': サイクルスチール転送(1ワード/1バイト転送毎に*Dreq 信号がアサート/ネゲートを繰り返す転送) '1': バースト転送(FIFO バッファレディで*Dreq 信号をアサートし、割り当てた FIFO バッファへのアクセス終了時に*Dreq 信号をネゲートする転送) ¹
転送方法 [DFORM]	Dn_FIFO データレジスタアクセスに使用される信号の選択 '00': *Dack 信号、及び、リード/ライト信号で制御 (チップセレクト信号、アドレス信号は don't care) '01': *Dack 信号のみで制御 (チップセレクト信号、アドレス信号、リード/ライト信号は don't care) '10': チップセレクト信号、アドレス信号、及びリード/ライト信号で制御 (*Dack 信号はインアクティブ固定にしてください) '11': (設定禁止)
DMA 割り込みモード [INTM]	バッファレディ割り込み発生条件の選択 ² '0': OUT バッファ設定時:DMA 転送終了 EPB_RDY ³ = '1'がセットされる(バッファレディ割り込み発生) IN バッファ設定時:EPB_RDY ビットは'1'にセットされない(バッファレディ割り込みは発生しない) '1': CPU_FIFO と同様の条件(FIFO バッファ読み出し可能/FIFO バッファ書き込み可能)でバッファレディ割り込みが発生する
DMA イネーブル [DMAEN]	*Dreq 信号出力許可/不許可の選択 ² '0': *Dreq 信号を出力しない '1': Dreq ⁴ の状態を*Dreq 信号に出力する

¹ バースト転送時の*Dreq 信号のネゲート条件は以下のとおりです。

- (1)非連続転送モード時:以下(A)~(C)のいずれかの場合
 - (A)すべての受信データ読み出し終了時
 - (B)最大パケットサイズ分のデータ書き込み終了時
 - (C)ショートパケット分のデータ書き込み(zero-length パケットの場合も含む)終了後に IVAL='1'をセット時
- (2)連続転送モード時:以下(A)~(C)のいずれかの場合
 - (A)受信の場合(ショートパケット受信時も含む)
 - ⇒FIFO バッファの片側のデータ読み出し終了時(ダブルバッファの場合は片側の読み出し終了時)
 - (B)[(送信データバイト数)≥FIFO バッファバイト数(ダブルバッファの場合は片側のバイト数)]の場合
 - ⇒割り当てた FIFO バッファ分のデータ書き込み終了時(ダブルバッファの場合は片側の読み出し終了時)
 - (C)[(送信データバイト数)<FIFO バッファバイト数(ダブルバッファの場合は片側のバイト数)]の場合
 - ⇒すべての送信データ書き込み終了後、IVAL='1'をセット時

² INTM='0'、かつ、DMAEN='0'の組み合わせでは使用しないでください。

³ 割り込みステータスレジスタ 1(Address 1Ah)

⁴ Dn_FIFO 制御レジスタ(Address 4Ah、52h)

表 2-6 DMA インターフェース用パラメータ有効組み合わせ一覧

No	B U S T	D F O R M	I N T M	D M A E N	備考
1	0	00	0	1	M66290A(1word 転送モード)互換モードです
2	0	00	1	1	組み合わせ No.1 で、バッファレディ毎にバッファレディ割り込みを発生させるモードです。
3	0	01	0	1	1 サイクル転送(*Dack 信号の立下りでソースからデータを読み出し、*Dack 信号の立ち上がりでディスティネーションにデータを書き込む)用のモードです。結線方法にご注意ください。
4	0	01	1	1	組み合わせ No.3 で、バッファレディ毎にバッファレディ割り込みを発生させるモードです。
5	0	10	0	1	*Dack ハンドシェイクを持たない DMA コントローラ用のモードです。 *Dack 信号はインアクティブ固定にしてください。
6	0	10	1	0	CPU アクセス用(DMA コントローラ不使用)のモードです。*Dreq 信号/*Dack 信号を使用しません。*Dack 信号はインアクティブ固定にしてください。
7	0	10	1	1	組み合わせ No.5 で、バッファレディ毎にバッファレディ割り込みを発生させるモードです。
8	1	00	0	1	組み合わせ No.1 のバースト転送モードです。
9	1	00	1	1	組み合わせ No.2 のバースト転送モードです。
10	1	01	0	1	組み合わせ No.3 のバースト転送モードです。
11	1	01	1	1	組み合わせ No.4 のバースト転送モードです。
12	1	10	0	1	組み合わせ No.5 のバースト転送モードです。
13	1	10	1	1	組み合わせ No.7 のバースト転送モードです。

*Dreq 信号は、DMA_EP[DMA 転送エンドポイント指定ビット]に指定されたエンドポイントのバッファがレディ状態になった時にアサートされます。ここで、バッファレディ状態とは、そのエンドポイントの転送方向が OUT(ホストからのデータを受信)に設定されていれば読み出し可能状態を指し、そのエンドポイントの転送方向が IN(ホストへデータを送信)に設定されていれば書き込み可能状態を指します。

DMA 転送では CPU アクセスと同様にエンドポイントに割り当てた FIFO バッファの状態によりバッファノットレディ割り込み、バッファエンpty/サイズオーバーエラー割り込みが発生します。しかし、バッファレディ割り込みの発生条件に関しては、表 2-5に示すとおり、INTM の設定によって差異があります。

INTM='0'の場合、バッファレディ割り込みの発生によりショートパケットを受信したことを認識することができます。DMA_DTLN¹を参照することにより、ショートパケットのバイト数、または、連続受信モードではショートパケット受信時の受信データのバイト数を示します。INTM='0'の設定で使用する場合、バッファレディ割り込み処理の中で必ず BCLR='1'を行い FIFO バッファをクリアしてください。

¹ Dn_FIFO 制御レジスタ(Address 4Ah、52h)

2.7.2. 16bit バス幅接続時に Dn_FIFO データレジスタで奇数バイト数データの読み出し/書き込みを行う方法

2.7.2.1. OUT 方向(読み出し)の場合

(2n+1)バイト受信する場合、DMA コントローラの転送カウンタを(2(n+1))バイト((n+1)回)に設定し、全データ読み出し後、(n+1)回目の転送データの上位を無視してください(n+1 回目のリードデータは上位/下位同じデータになります)。

この場合はショートパケット受信時ですので、バッファレディ割り込み処理の中で上記判定を行います。読み出した FIFO バッファの受信データ数を DMA_DTLN で確認してください。

2.7.2.2. IN 方向(書き込み)の場合

(2n+1)バイト送信する場合、DMA コントローラの転送カウンタを(2n)バイト(n 回)に設定し、DMA コントローラの終了割り込みの中で、Octl¹=1¹、または、EPI_Octl=1²を設定後、最後の 1 バイトを書き込み書きこみ、IVAL=1¹をセットしてください。

2.7.3. Dn_FIFO レジスタ経由で FIFO バッファにショートパケットデータを書き込む方法

DMA コントローラの転送カウンターに書き込むバイト数を設定し、DMA コントローラの終了割り込み処理にて、IVAL=1¹をセットしてください。

2.7.4. 連続転送モード時に、FIFO バッファ容量未滿かつ最大パケットサイズの整数倍のデータを読み出す方法

連続転送モード時に、Dn_FIFO データレジスタ経由で、FIFO バッファ容量未滿かつ最大パケットサイズの整数倍のデータを読み出す場合の手順例を示します。

USB specification revision1.1 の 5.8.3 章で、bulk 転送のデータ転送完了は次のように定義されており、ここでは、前者の場合(デバイス側が受信データ数を予測可能な場合)について例を示します。

- Has transferred exactly the amount of data expected
- Transfers a packet with a payload size less than *wMaxPacketSize* or transfers a zero-length packet.

この場合には、トランザクション回数機能が使用できます。データ受信前に TRNCNT³にトランザクション回数を設定してください。

M66291 は設定した回数分のパケット受信後、*Dreq 信号をアサートします。手順としては、次のとおりです。

- (1) 受信データ数がわかるまで、PID²=00'(NAK)にしておく
- (2) TRCLR⁴=1' (トランザクションカウンタクリア)
- (3) TRNCNT=(受信パケット数) (同時に DMA コントローラの転送カウンタセット)
- (4) TREN⁴=1' (トランザクションカウント有効)
- (5) PID=01'(Buf) (受信開始)
- (6) (受信、DMAC によるデータ読み出し)
- (7) 受信完了時(DMA コントローラ割り込み処理の中)、PID=00'(NAK)

¹ Dn_FIFO 選択レジスタ(Address 48h、50h)

² EPI コンフィギュレーションレジスタ 1(Address 62h、66h、6Ah、6Eh、72h、76h)

³ DMAn_トランザクション回数レジスタ(Address 4Eh、56h)

⁴ Dn_FIFO 制御レジスタ(Address 4Ah、52h)

3. 初期設定

USB 通信用の初期設定は、図 2-5 中「USB 初期設定」に示したフローとなります。

3.1. M66291 起動

M66291 の発振が安定してから初期化処理を行います。PLL 動作許可設定後はクロックによる不安定な動作を避けるために待ち時間(1ms 以上)が必要です。TrON ポートを”H”出力することにより D+信号がプルアップされ、ホスト(アップストリーム側)がデバイス接続を検知します。

S/W リセット(USBE¹=0)実行時の各レジスタ状態は M66291 データシート【レジスタ一覧】を参照ください。

M66291 起動を図 3-1に示します。図 3-1に関する補足事項は以下のとおりです。

- (1) 水晶発振子により発振供給する場合の起動例を示しています。
- (2) ウェイト後の SW リセットはホストに対する非接続通知も兼ねており、USB バスリセット信号を受信するために行っています。
- (3) ホストより Vbus が供給されている場合に行う M66291 起動(初期設定)時の設定例です。

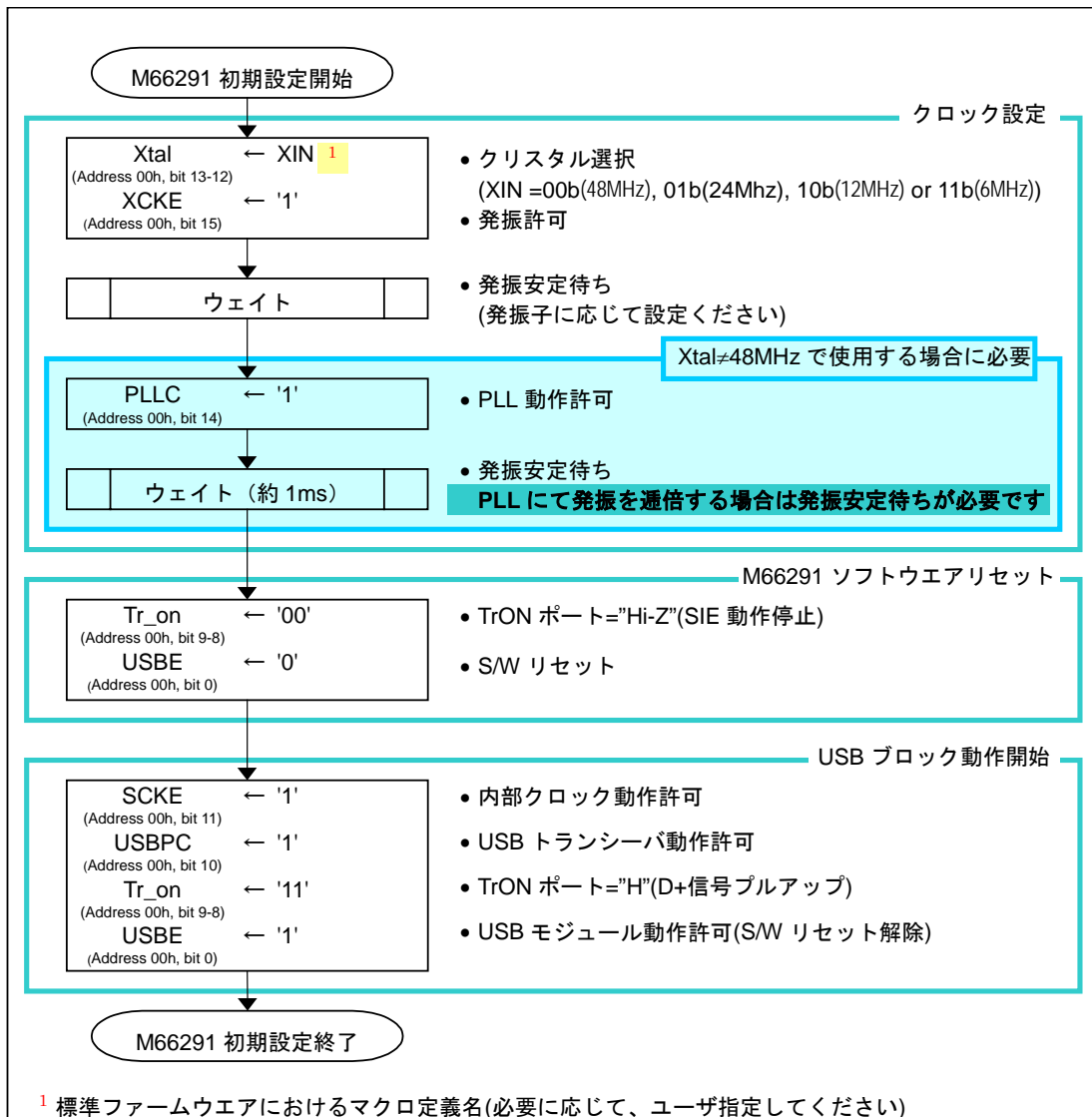


図 3-1 M66291 起動例

¹ USB 動作許可レジスタ(Address 00h)

3.2. CPU 接続設定

M66291 には、制御 CPU との接続性向上のためにいくつかの機能があります。M66291 初期設定後に、必要に応じて以下の設定を行ってください。M66290A 互換設定の場合、デフォルト設定で使用される場合はこれらの処理は必要ありません。

- (1) 割り込み端子*INT0、*INT1 の極性("L"アクティブ(デフォルト)/"H"アクティブ)、及び、割り込みセンス(エッジセンス(デフォルト)/レベルセンス)設定
- (2) *Dreq0、*Dack0、*Dreq1、*Dack1 の極性設定(DMA インターフェース使用時)("L"アクティブ(デフォルト)/"H"アクティブ)
- (3) 各割り込み要因の出力端子アサイン(デフォルトではすべて*INT0 にアサイン)

上記(1)~(3)の処理は S/W リセット解除後(USBE='0'→'1')に行ってください。S/W リセット実行直後はデフォルト設定に戻りますので、S/W リセット解除後に必ず再設定を行ってください。

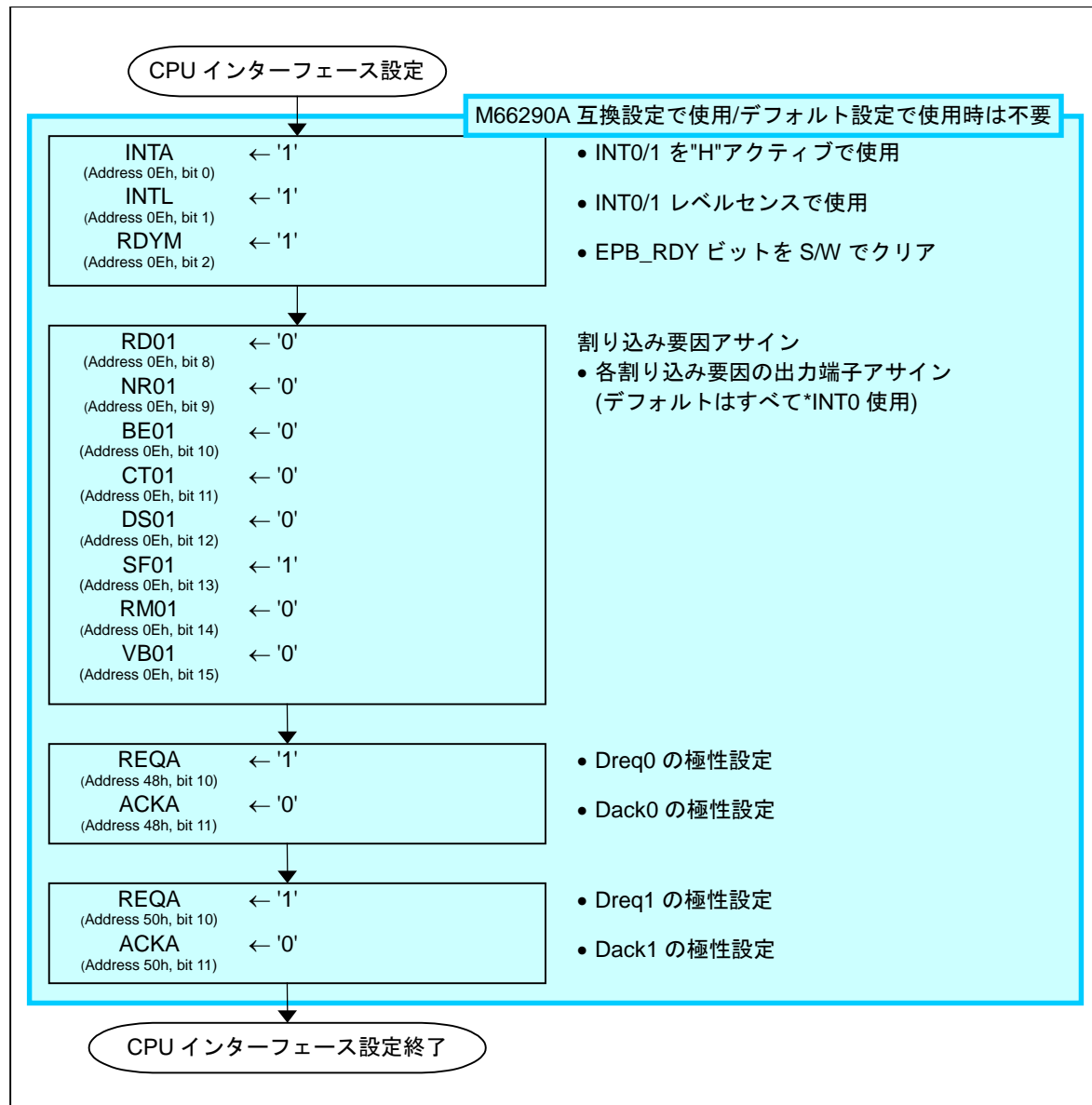


図 3-2 CPU 接続設定例

3.3. コントロール転送準備(エンドポイント 0 初期設定)

M66291 の初期設定後に、コントロール転送を行える状態にするため、エンドポイント 0 初期設定を行ってください。エンドポイント 0 初期設定手順例を図 3-3に示します。図 3-3に関する補足事項は以下のとおりです。

- (1) エンドポイント 0 初期設定は、M66291 起動、CPU 接続設定後に行ってください。
- (2) FIFO バッファ先頭番号の指定は、送受信の FIFO バッファ領域が重ならないように行ってください。また、同時使用の可能性のあるエンドポイント 1-エンドポイント 6 の FIFO バッファ領域とも重ならないように指定してください。
コントロール転送が FIFO バッファを使用するのは Data ステージにおけるデータパケット送受信時です。Setup ステージのデータパケットの受信は専用レジスタに格納されます。
- (3) 旧タイプのホストコントローラ、及び、旧タイプのホスト OS に対応するため、最大パケットサイズは”8 バイト”を推奨しています。
- (4) 自動応答が有効な状態遷移は [4 コントロール転送](#)を参照ください。
- (5) FIFO バッファクリアは $IVAL^1 = '0'$ 、 $BLCR^1 = '1'$ を同時に書き込んでください。
- (6) 図 3-3のエンドポイント 0 の各設定はマクロ定義されている場合の例を示しています。

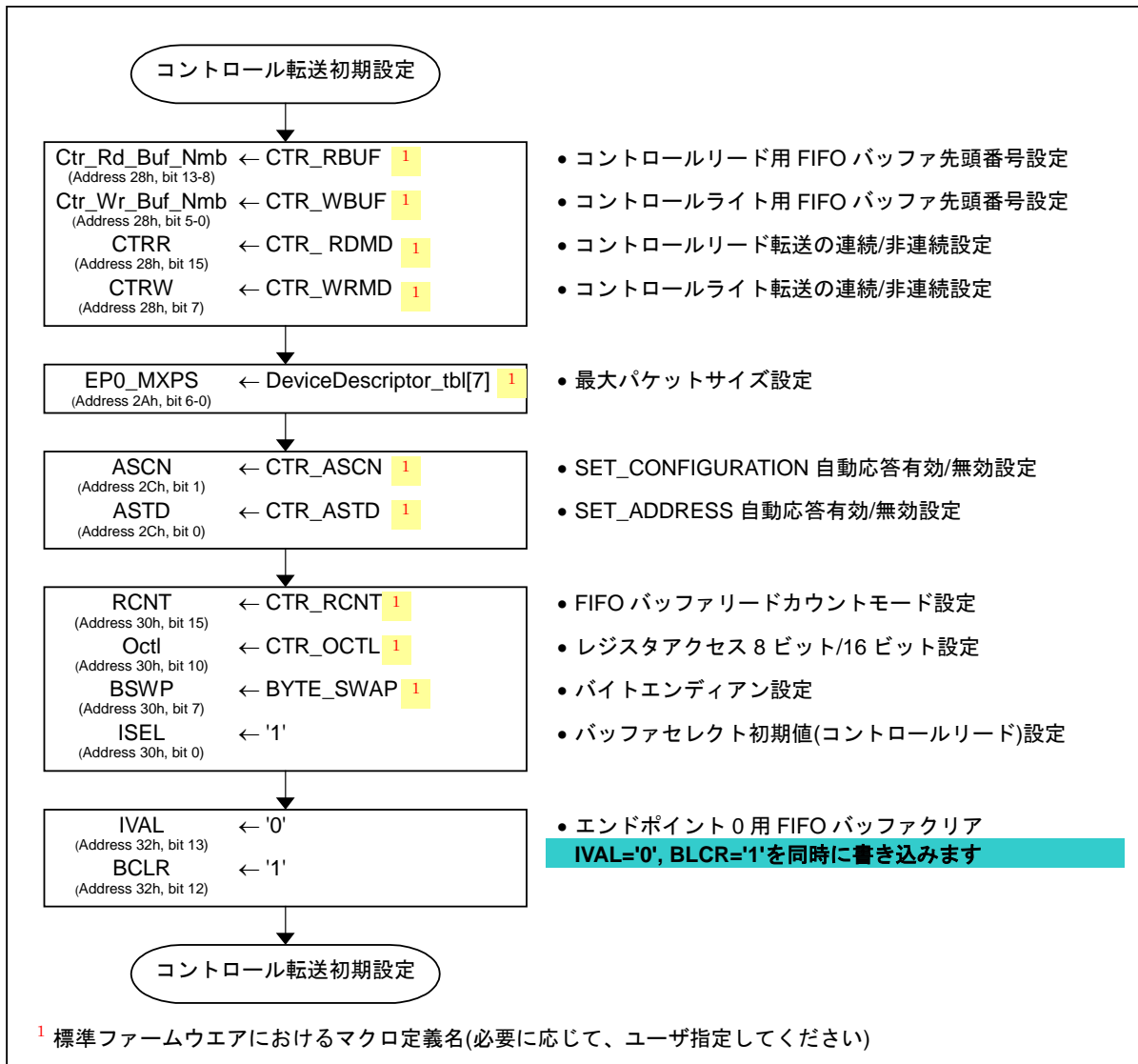


図 3-3 エンドポイント 0 初期設定手順例

¹ EP0_FIFO 制御レジスタ(Address 32h)

3.4. USB 初期割り込み要因許可

M66291 により USB 通信を開始するために、M66291 の割り込み許可設定を行ってください。割り込み許可に関する主な注意点を以下に示します。

- (1) M66291 の割り込み許可設定は M66291 起動、エンドポイント 0 初期設定処理後に行ってください。
- (2) 制御用 CPU 及び他チップと、M66291 割り込み許可に関するタイミングは、システム構成及びご用途に合わせて設計してください。
- (3) エンドポイント 1-エンドポイント 6 に対するデータ転送割り込み許可(バッファレディ割り込み、バッファノットレディ割り込み、バッファEMPTY/サイズオーバーエラー割り込み)は、Set Configuration リクエスト受信後に行ってください。

Vbus 供給されている(USB デバイスがすでに PC に接続されている)場合の USB 初期割り込み要因許可設定例を図 3-4に示します。図 3-4の補足説明は以下のとおりです。

- (1) VBUS 割り込みを使用しない設定例です。

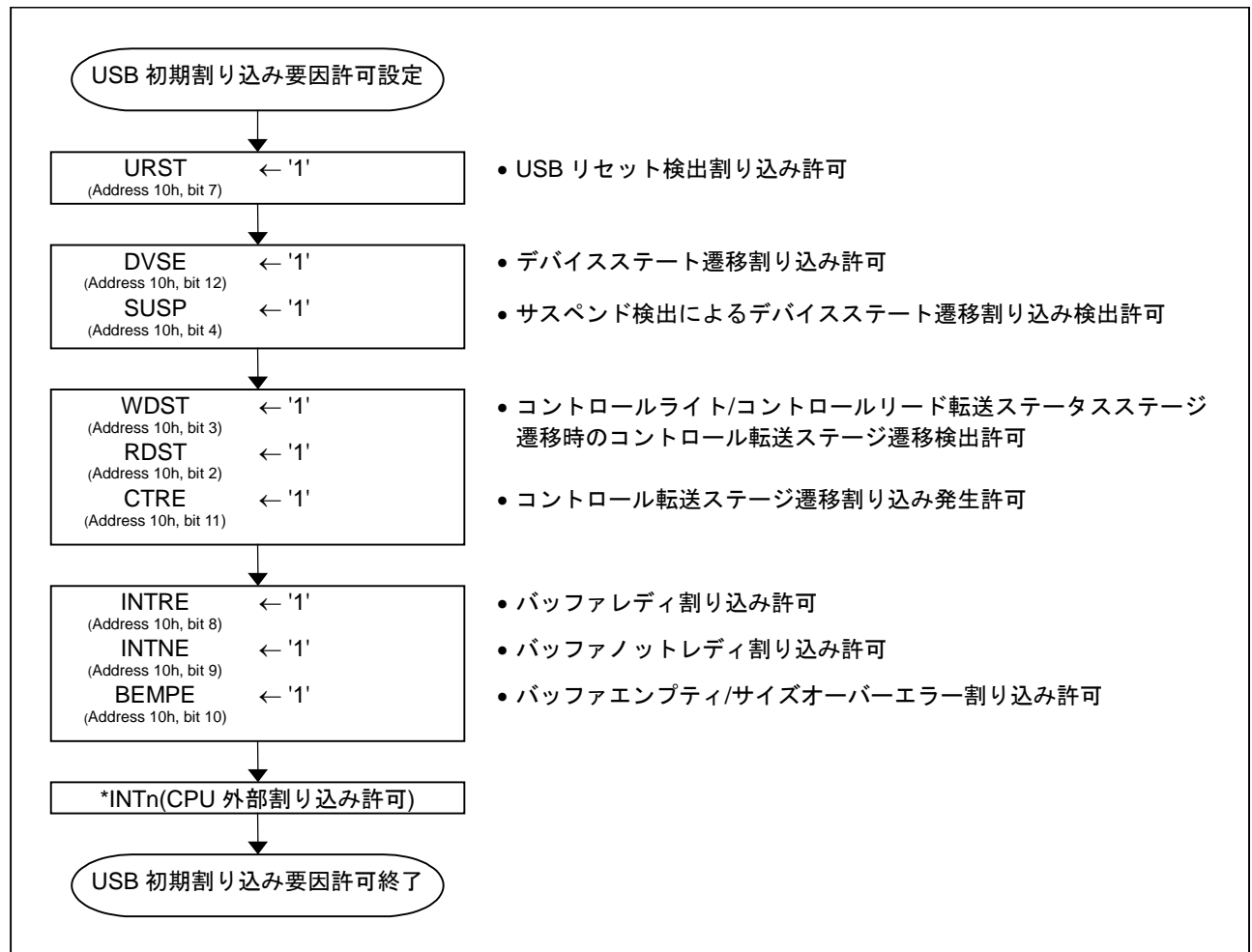


図 3-4 USB 初期割り込み要因許可設定例

4. コントロール転送

4.1. コントロール転送概要

コントロール転送は主にセットアップ時に使用される双方向転送で、少なくとも2つのトランザクションステージ(セットアップステージ、ステータスステージ)により構成されます。また、セットアップステージとステータスステージ間に、データステージが存在する場合があります。

USB 通信では、エンドポイント 0 によるコントロール転送及びエンドポイント 0 のコントロール転送によりホストからデバイスに送信されるいくつかのリクエストに対する応答は必須機能です。

M66291 は、エンドポイント 0 によるコントロール転送及び、セットアップ処理(バスエニュメレーション)制御用プログラムが簡単に記述できるように以下の4機能を備えています。

- | | |
|--------------------|----------------|
| (1) デバイスステート遷移管理機能 | 【本資料4.2、4.3参照】 |
| (2) データステージ遷移管理機能 | 【本資料4.4、4.5参照】 |
| (3) 連続送受信機能 | 【本資料4.7、4.9参照】 |
| (4) 自動応答機能 | 【本資料4.9参照】 |

4.1.1. コントロール転送を構成するトランザクション

コントロール転送は、以下3種類のトランザクションで構成されます。

(1) セットアップステージ:

セットアップ、データ(DATA0)、及び、ハンドシェイクパケットにより構成されます。

SETUP トークン → DATA0(リクエスト) → ACK

本資料では、この一連のトランザクションを **SETUP(DATA0)** と表記します。

(2) データステージ:

バルク転送と同様にデータトランザクションが必要なだけ繰り返されます。

①コントロールリードの場合

IN トークン → DATA(1/0) → ACK IN トークン → DATA(0/1) → ACK ... IN トークン → DATA(1/0) → ACK
 (1 トランザクション) (1 トランザクション) (1 トランザクション)

②コントロールライトの場合

OUT トークン → DATA(1/0) → ACK OUT トークン → DATA(0/1) → ACK ... OUT トークン → DATA(1/0) → ACK
 (1 トランザクション) (1 トランザクション) (1 トランザクション)

本資料では、この一連のトランザクションを **IN/OUT(DATA)** と表記します。

(3) ステータスステージ:

データステージと逆方向のトークン、及び、zero-lengthパケット(DATA1)で構成されます。

(A) コントロールリードの場合

OUT トークン → DATA1(zero-length パケット) → ACK

(B) コントロールライトの場合

IN トークン → DATA1(zero-length パケット) → ACK

本資料では、この一連のトランザクションを **IN/OUT(DATA1:HSK)** と表記します。

4.1.2. コントロール転送の種類

コントロール転送には、以下の 3 種類があります。

(1) コントロールリード転送:



(2) コントロールライト転送:



(3) ノーデータコントロール転送:



M66291 はステージ管理機能を備えており、コントロール転送遷移割り込み(CTRTR=1')によりステージ遷移を通知します。CPU は割り込み要因によってコントロール転送の該当するステージ処理を行います。セットアップステージで取得されたデータの内容によって、デバイスはホストの要求するリクエスト種類が明らかになり、該当するコントロール転送の処理を行うことができます。

また、M66291 は連続送受信機能を備えています。コントロール転送における連続送受信機能は、データステージのデータを最大 256 バイトまで割り込みを発生せずに、連続してデータを送受信する機能です。連続送受信機能を用いることで、データステージでは 1 トランザクションごとに FIFO アクセスを行う必要がなくなり、CPU 稼働率を高めることができます。

4.2. USB デバイスステート遷移

M66291 は USB デバイスステートの Powered、Default、Address、Configured、Suspended ステートのデバイスステート遷移管理を H/W で行います。レジューム検出以外のデバイスステート遷移は、デバイスステート遷移割り込み(DVST¹=1)によって CPU へ通知します。レジューム検出によるデバイスステート遷移は、レジューム割り込み(RESM¹=1)によって CPU へ通知します。

自動応答設定時は Set Address、Set Configuration リクエストに対して、CPU への割り込みを発生させずに転送を完了させることができます。

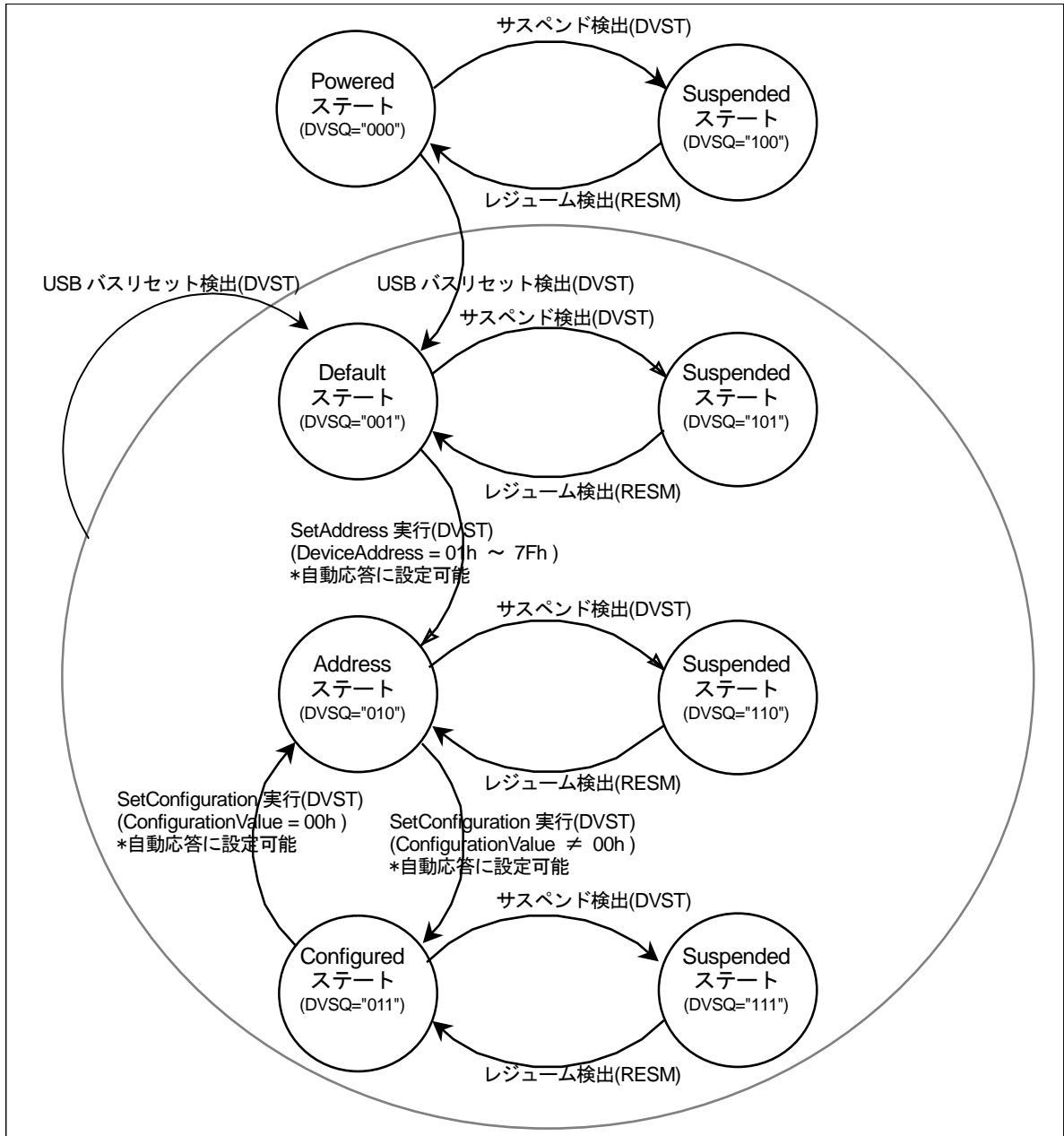


図 4-1 デバイスステート遷移

¹ 割り込みステータスレジスタ 0(Address 18h)

4.3. デバイスステート遷移処理

デバイスステート遷移が正常に実行されると、M66291 は DVSQ[デバイスステート]¹に遷移後のデバイスステート値をストアし、デバイスステート遷移割り込みを発生(DVST='1')させます。デバイスステート遷移割り込みの要因には以下の 4 種類があり、各割り込み要因毎にデバイスステート遷移割り込み発生時のマスク(各要因検出時の DVST ビットセットの禁止/許可)を設定することができます。

デバイスステート遷移割り込みの発生要因

- (1) USB リセット検出(許可ビット:URST¹)
- (2) Set Address 実行(許可ビット:SADR¹)
- (3) Set Configuration 実行(許可ビット:SCFG¹)
- (4) サスペンド検出(許可ビット:SUSP¹)

制御プログラムは、割り込みによって検出したデバイスステート遷移に必要な処理を実行し、ステート遷移に伴うユーザプログラムを実行します。

各デバイスステートへの遷移条件と、遷移時に必要な処理を以下に示します。

表 4-1 デバイスステート状態遷移条件と処理

デバイスステート	遷移条件	遷移時の処理
パワーステート (Powered state)	S/W リセット	図 3-2 CPU 接続設定例 を参考に必要な処理を行ってください
デフォルトステート (Default state)	USB バスリセット検出	(1)エンドポイント 1-エンドポイント 6 に対するデータ転送割り込み禁止/割 り込みステータスクリア (2)エンドポイント 1-エンドポイント 6 に割り付けた FIFO バッファクリア
アドレスステート (Address state)	(1)デフォルトステートにおいて、 Set Address リクエスト検出 (2)構成ステートにおいて、 configuration value =0 の Set Configuration リクエスト検出	アドレス保持等
構成ステート (Configured state)	configuration value ≠ 0 の Set Configuration リクエスト検出	構成番号保持等
サスペンドステート (Suspended state)	Suspend 信号検出	サスペンド処理

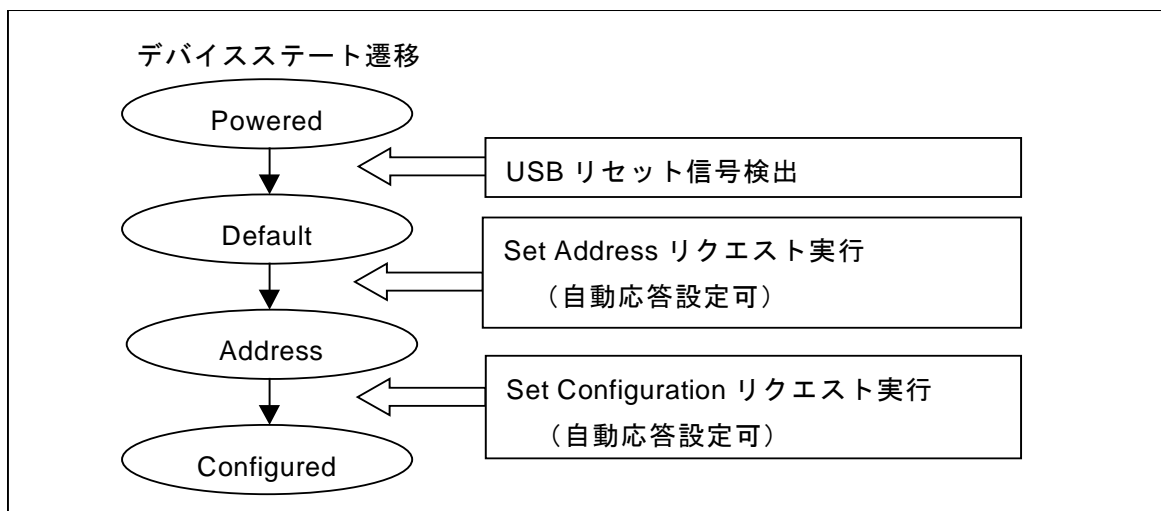


図 4-2 デバイスステート状態遷移

¹ 割り込み許可レジスタ 0(Address 10h)

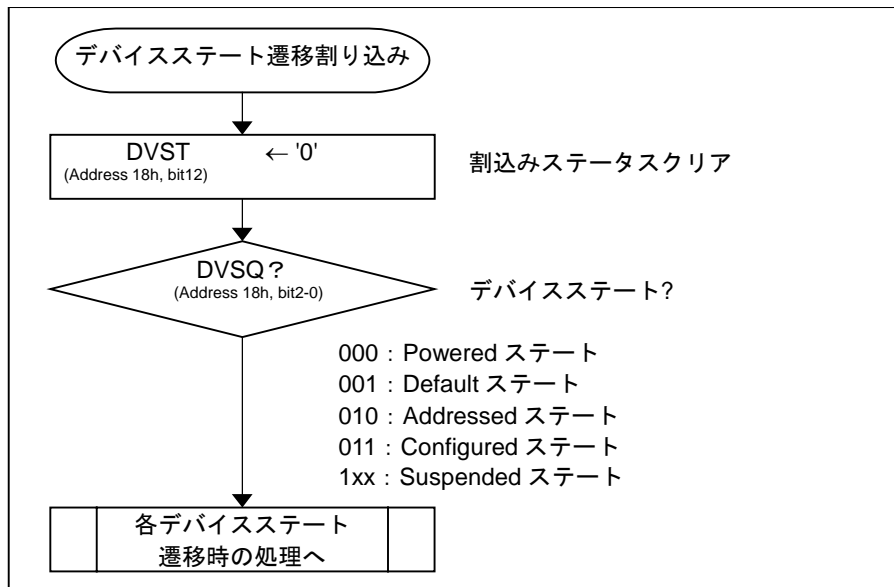


図 4-3 セットアップステージ制御手順例

4.4. コントロール転送ステージ遷移

M66291 はコントロール転送ステージのセットアップステージ、データステージ、ステータスステージのステージ遷移管理を H/W で行います。M66291 はコントロール転送ステージ遷移を検出すると、CTSQ[コントロール転送ステージ]¹に遷移後のコントロール転送ステージ値をストアし、コントロール転送ステージ遷移割り込み(CTRT¹=1')を発生させます。

自動応答設定時は Set Address、Set Configuration リクエストに対しては、CPU に割り込みを発生させずに転送を完了させることができます。

コントロール転送ステージ遷移割り込みの要因としては以下の 5 種類があり、セットアップステージ完了以外の要因については各割り込み要因毎にコントロール転送ステージ遷移割り込み発生マスク(各要因検出時の CTRT ビットセットの禁止/許可)を設定することができます。

コントロール転送ステージ遷移割り込みの発生要因

- (1) セットアップステージ完了(マスク不可)
- (2) コントロールライト転送ステータスステージ遷移(許可ビット:WDST¹)
- (3) コントロールリード転送ステータスステージ遷移(許可ビット:RDST¹)
- (4) コントロール転送完了(許可ビット:CMPL¹)
- (5) コントロール転送シーケンスエラー(許可ビット:SERR¹)

H/W で検出可能なコントロール転送シーケンスエラーは次に示す 7 つのエラーです。

- (1) コントロールライトデータステージでの IN トークンパケット受信
(データステージの OUT トークンパケットに対して 1 度も ACK 応答をしていない状態での IN トークンパケット受信)
- (2) コントロールライトステータスステージでの OUT トークンパケット受信
- (3) コントロールリードデータステージでの OUT トークンパケット受信
(データステージの IN トークンパケットに対して 1 度もデータ転送をしていない状態での OUT トークンパケット受信)
- (4) コントロールリードステータスステージでの IN トークンパケット受信
- (5) コントロールリードステータスステージでの zero-length パケット以外のデータパケット受信
- (6) ノーデータコントロールステータスステージでの OUT トークンパケット受信
- (7) 最大パケットサイズを超えたデータの受信

コントロールライトデータステージにおいて、リクエストの wLength 値を超えたデータパケットを受信した場合は、シーケンスエラーと認識することはできません。

なお、H/W でコントロール転送シーケンスエラーを検出した場合には、自動的に応答 PID は STALL('1x')になり、次のセットアップパケットを受信するまで PID='1x'の状態を保持します。

図 4-4にコントロール転送ステージ遷移を示します。

¹ 割り込み許可レジスタ 0(Address 10h)

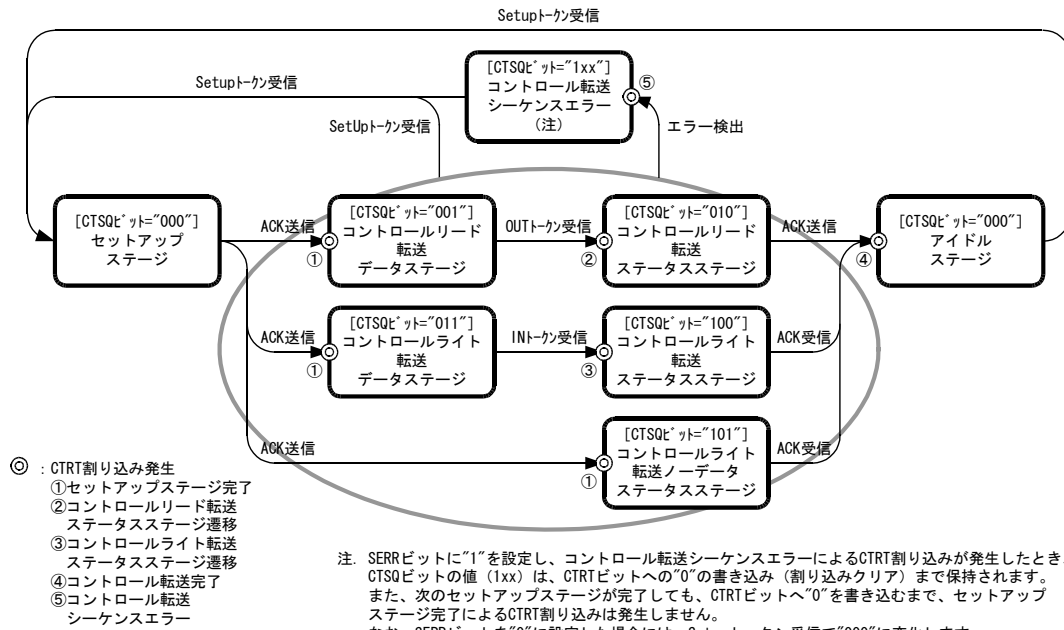


図 4-4 コントロール転送ステージ遷移

4.5. コントロール転送ステージ遷移処理

コントロール転送ステージ遷移割り込み(CTRT='1')により、M66291 のコントロール転送ステージ遷移を検出した CPU は、CTSQ[コントロール転送ステージ]を確認し該当する処理を行います。

M66291 は、セットアップステージのデータパケット(ホスト PC からのリクエスト)を専用レジスタに格納しています。また、データステージのデータパケットは、エンドポイント 0 用 FIFO バッファを使用します。

コントロール転送の転送ステージと遷移時の処理を表 4-2 に示します。

表 4-2 コントロール転送の転送ステージと遷移時の処理

転送ステージ	遷移時の処理
アイドルまたはセットアップステージ	特に処理は不要(ユーザの必要に応じて処理を行ってください)
コントロールリード転送データステージ	リクエストデータ解析 データ送信準備(含む送信データ書き込み)
コントロールライト転送データステージ	リクエストデータ解析 データ受信準備 受信データ読み出し
ノーデータコントロール転送ステータスステージ	リクエストデータ解析
コントロールリード転送ステータスステージ	特に処理は不要(ユーザの必要に応じて処理を行ってください)
コントロールライト転送ステータスステージ	特に処理は不要(ユーザの必要に応じて処理を行ってください)
コントロール転送シーケンスエラー	セットアップ再受信準備

コントロールライト転送において、最終パケットを受信したことによるバッファレディ割り込みとステータスステージ遷移割り込みがほぼ同時に発生する場合がありますので、割り込み処理プログラムは、コントロール転送ステージ遷移割り込み処理の前にエンドポイント 0 バッファレディ割り込み処理を行うように設計してください。

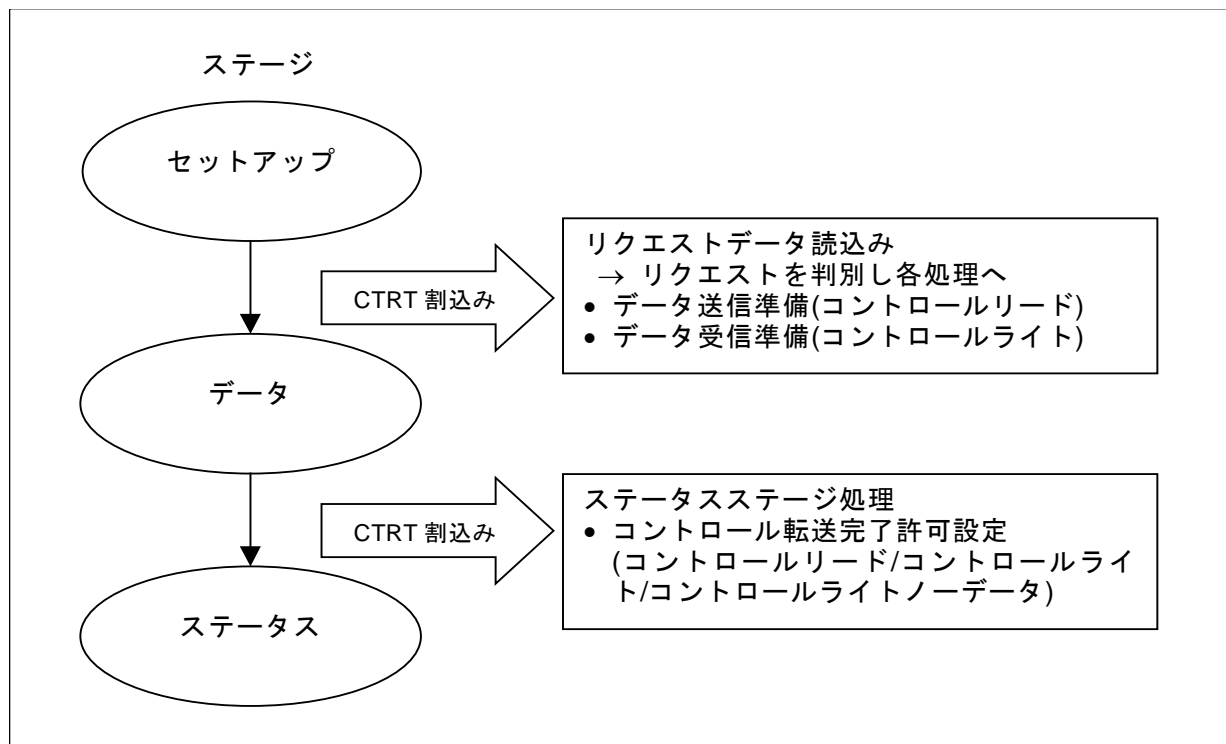


図 4-5 ステータスステージ遷移処理

4.6. セットアップステージ

セットアップトランザクションにて、ホストから 8 バイトのリクエスト(セットアップステージのデータパケット)が転送され、以下の 4 レジスタに格納されます。

表 4-3 リクエスト格納レジスタ

レジスタ名	アドレス	ビット	格納値
Request	20h	15-8	bRequest
		7-0	bmRequest
Value	22h	15-0	wValue
Index	24h	15-0	wIndex
Length	26h	15-0	wLength

リクエストをレジスタに格納した後、M66291 は VALID[セットアップパケット検出ビット]¹='1' をセットし、CTSQ[コントロール転送ステージ]¹にコントロール転送ステージ管理機能により該当する値を設定し、コントロール転送ステージ遷移割り込み(CTR¹'=1')を発生させます。

制御プログラムは、VALID='0'を設定したのちに、リクエストをレジスタから読み出し、エラー判定、リクエスト解析処理を施し、データステージの準備を行います。あるリクエストのデータステージ処理中、あるいはステータスステージ処理中に新たなセットアップトランザクションが発生した場合には、それまでのリクエスト処理を中断し、新たなリクエスト処理を行なう必要があります。このため、コントロール転送ステージ遷移割り込み処理の中では毎回必ず VALID の値を確認してください。

M66291 はセットアップトランザクションを正常受信すると自動的に”ACK”応答します。また、セットアップパケット受信後にエンドポイント 0 の PID を自動的に”NAK”設定します。

コントロール転送遷移割り込み処理例を図 4-6に示します。

¹ 割り込みステータスレジスタ 0 (Address 18h)

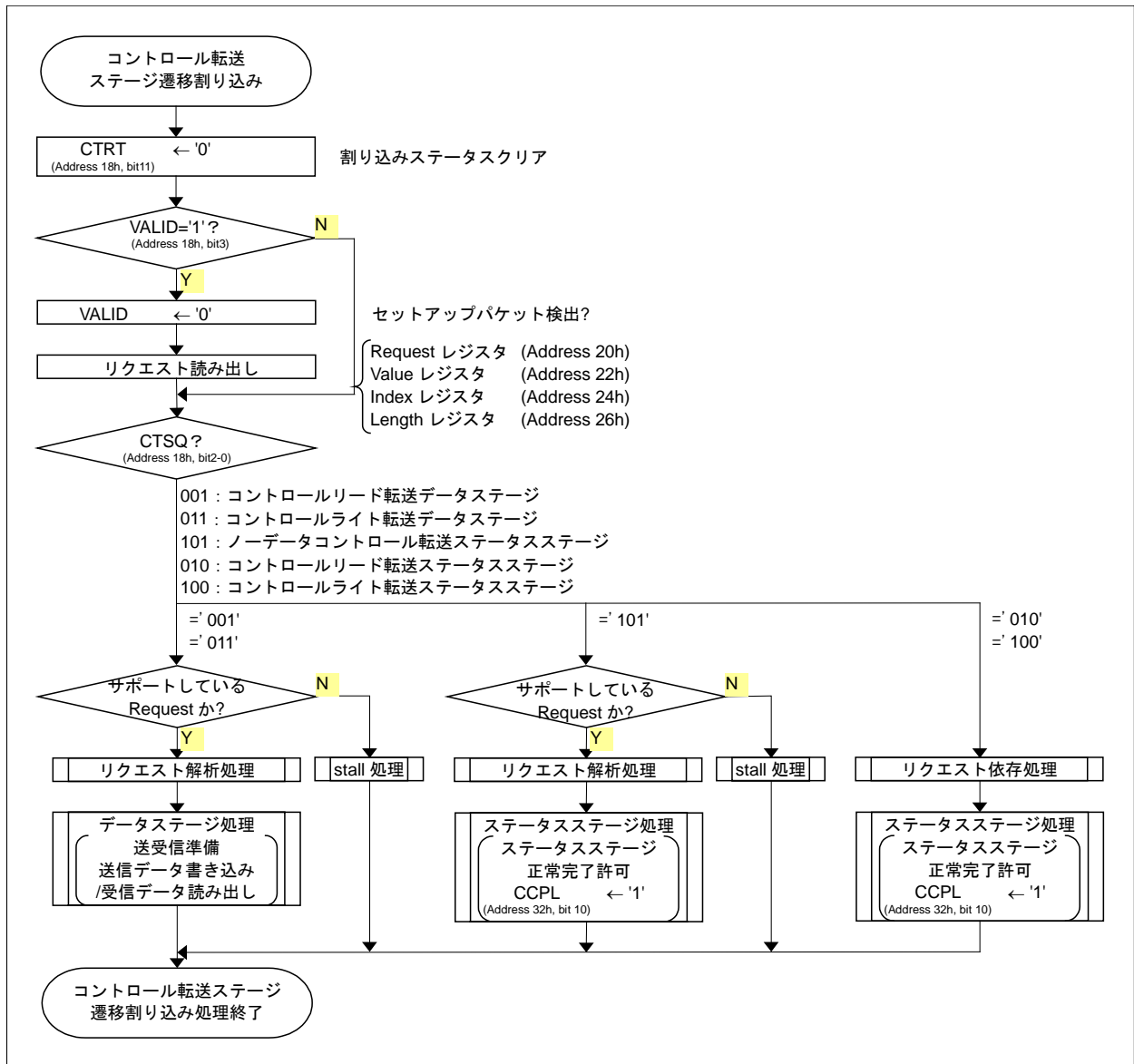


図 4-6 コントロール転送遷移割り込み処理例

4.7. コントロールリード転送データステージ処理例

セットアップステージにて受信したリクエストがコントロールリードリクエストの場合は、データステージにおいてホストPCに対してリクエストされたデータ送信を行ってください。

M66291 はコントロールリード時のデータ送信(データステージ)において連続転送機能を備えており、設定を行えば、複数パケットの連続送信が可能になります。設定可能な **SDLN[コントロールリード連続送信データ長]**¹の最大値は 256 バイトです。

連続送信設定の場合で、FIFO バッファに書き込んだデータ長が最大パケットサイズで割り切れ(最大パケットサイズの n 倍)、かつ、256 バイト未満の場合は、全データ送出後に自動的に zero-length パケット (Null パケット)送出の準備を行います。但し、FIFO バッファに書き込んだデータ長が 256 バイトの場合は、自動的な zero-length パケット送出準備は行いませんので、制御プログラムで送出準備を行う必要があります。非連続送信設定の場合に、最後のデータパケットが最大パケットサイズの場合にも、制御プログラムで zero-length パケット送出準備を行う必要があります。²

ショートパケット送信の場合など、FIFO 容量未満のデータ送信時は、最終データ書き込み後に **IVAL='1'** 設定を行いパケット送出準備を完了してください。M66291 はデータステージで **OUT** トークンを受信すると、自動的にステータスステージにステージ遷移します。

非連続転送モード時コントロールリード転送処理例を図 4-7に、連続転送モード時コントロールリード転送処理例を図 4-8に示します。

¹ EP0 連続送信データ長レジスタ (Address 36h)

² USB specification 1.1 5.2.2 章に次のような規定があるためです。

When all of the data structure is returned to the host, the function should indicate that the Data stage is ended by returning a packet that is shorter than the *MaxPacketSize* for the pipe. If the data structure is an exact multiple of *wMaxPacketSize* for the pipe, the function will return a zero-length packet to indicate the end of the Data stage.

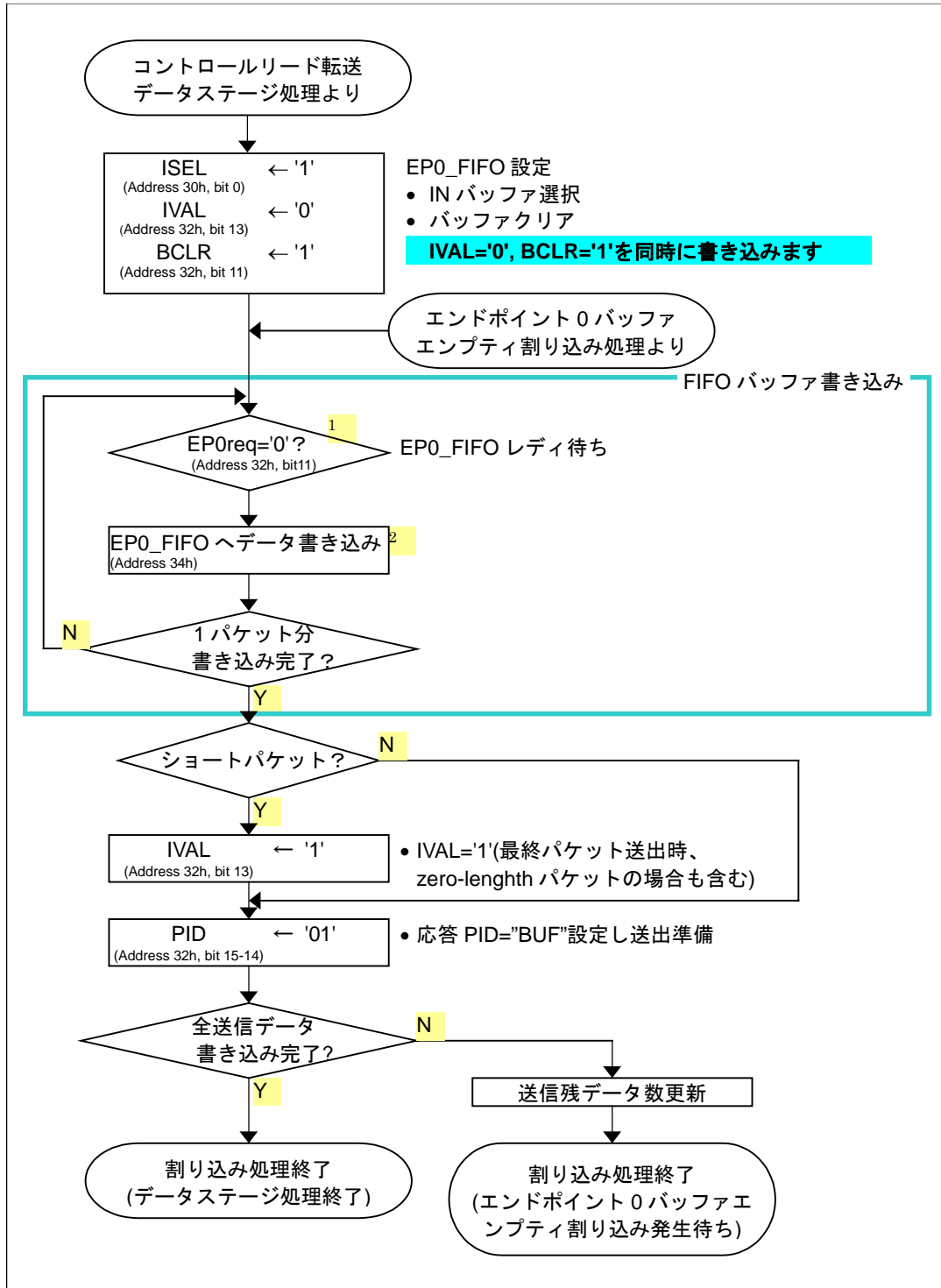


図 4-7 非連続転送モード時コントロールリード転送処理例

¹ FIFO データレジスタへのアクセスサイクルがデータシートを満たす場合は、この確認は省略可能です。

² 16bit バス幅接続のシステムで、奇数バイト数のデータ書き込みを行う場合は、16ビットモードで最後の1バイト直前まで書き込みし、Octl='1'設定により16bit→8bitに切り替えて最後の1バイトの書き込みを行います。16ビットモード時にバイトスワップ機能(BSWP='1')を使用する場合は、BSWP='0'に戻してからOctl='1'設定を行ってください。

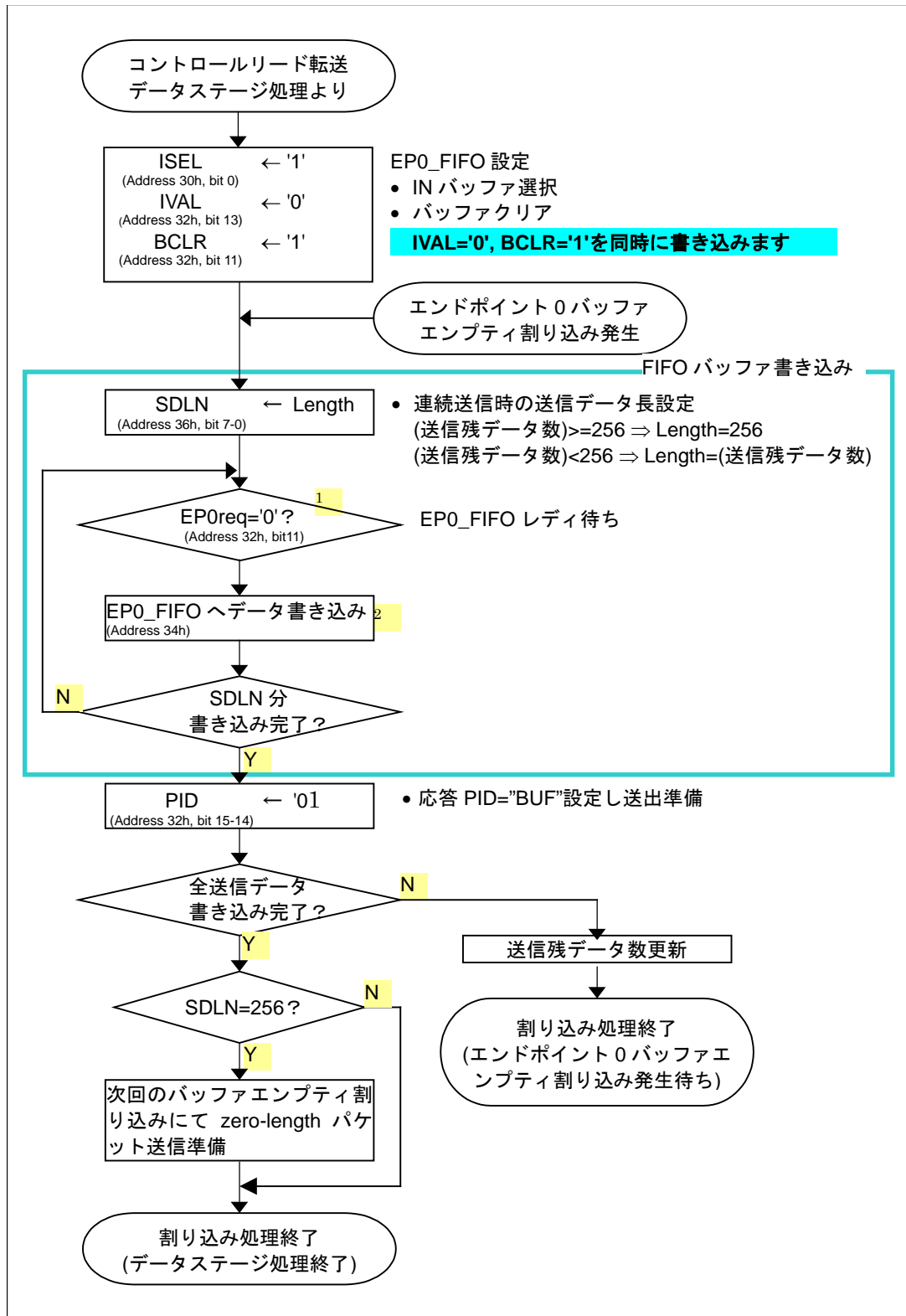


図 4-8 連続転送モード時コントロールリード転送処理例

¹ FIFO データレジスタへのアクセスサイクルがデータシートを満たす場合は、この確認は省略可能です。

² 16bit バス幅接続のシステムで、奇数バイト数のデータ書き込みを行う場合は、16ビットモードで最後の1バイト直前まで書き込みし、Octl='1'設定により16bit→8bitに切り替えて最後の1バイトの書き込みを行います。16ビットモードにバイトスワップ機能(BSWP='1')を使用する場合は、BSWP='0'に戻してからOctl='1'設定を行ってください。

4.8. コントロールライト転送データステージ処理例

セットアップステージにて受信したリクエストがコントロールライトリクエストの場合は、データステージにおいてホスト PC よりリクエストされたデータ受信を行ってください。

M66291 はコントロールライト時のデータ受信(データステージ)において連続転送機能を備えており、設定を行えば、複数パケットの連続受信が可能になります。コントロール転送の連続受信時に設定できるバッファ領域は最大 256 バイトです。256 バイトを超えるデータを受信する場合は複数回にわけて受信・読み出しを行ってください。

M66291 はデータステージで IN トークンを受信すると、自動的にステータスステージにステージ遷移します。

コントロールライト転送処理例を、図 4-9に示します。

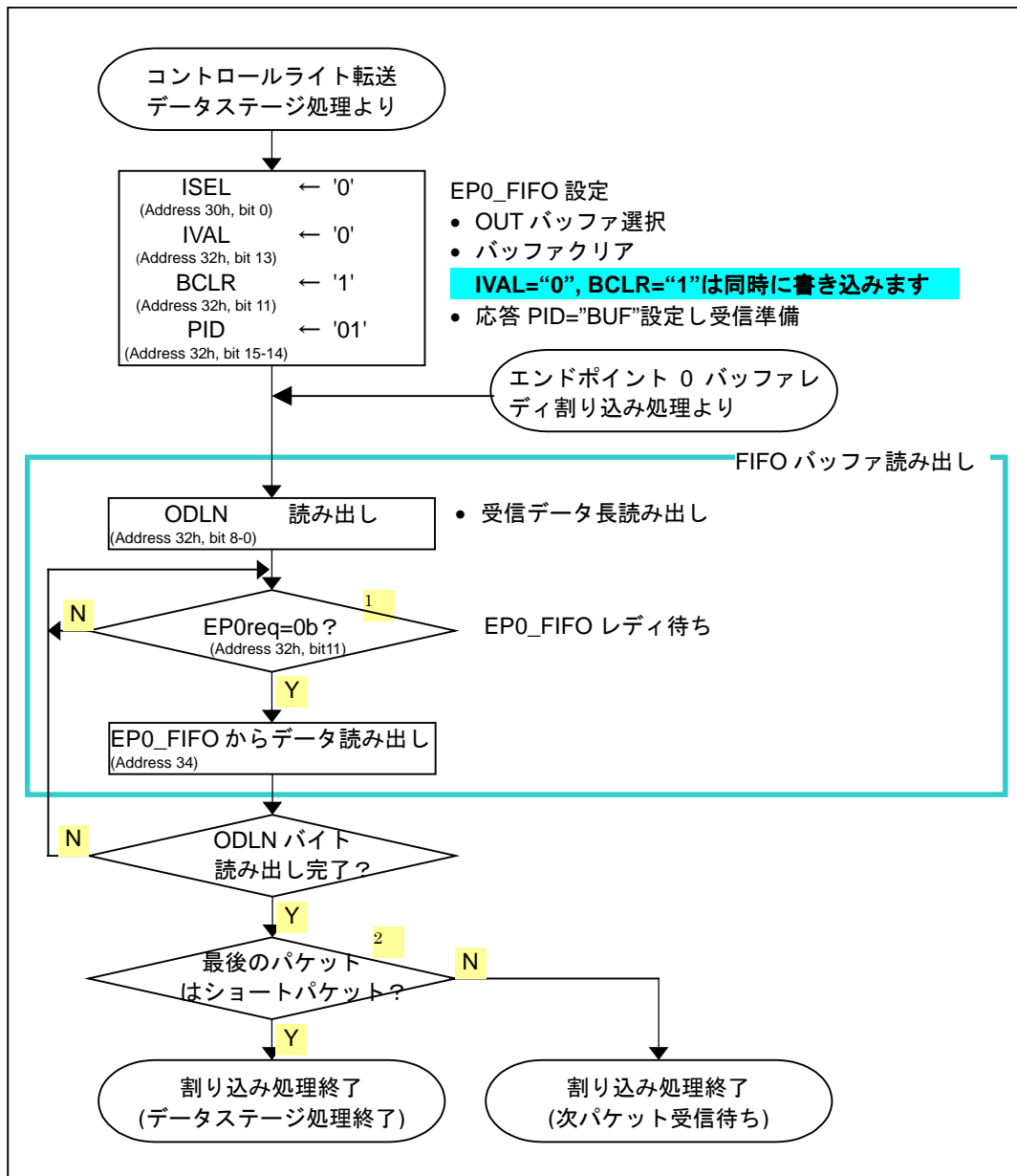


図 4-9 コントロールライト転送処理例

¹ FIFO データレジスタへのアクセスサイクルがデータシートを満たす場合は、この確認は省略可能です。

² 16bit バス接続のシステムで ODLN が奇数の場合は、最後の 1 ワード(2 バイト)を読み出し後、上位 8 ビットを破棄してください。

4.9. ノーデータコントロール転送処理(ステータスステージ)例

セットアップステージにて受信したリクエストがノーデータコントロール転送のリクエストの場合は、セットアップステージ以降はホストPCとのデータ送受信は行いません。

M66291 はセットアップステージでノーデータコントロール転送のリクエストを受信すると、自動的にステータスステージにステージ遷移します。なお、ノーデータコントロール転送リクエストの判定には、`bmRequestType` 及び `wLength` フィールドを使用します。

M66291 はノーデータコントロール転送である標準リクエスト `Set Address`、`Set Configuration` に対して、CPU に割り込みを発生させずに転送を完了させることができます(自動応答機能)。

M66291 が `Set Address` 自動応答を行うのは以下の条件を満たす場合のみです。その他のステートの `Set Address` リクエストに対しては自動応答を行いません。自動応答を行わない場合でも、M66291 は `USB アドレスレジスタ(Address 08h)` の更新を行います。

(1) デバイスステートがデフォルトステートで、`Set Address` リクエストを受信した。

M66291 が `Set Configuration` 自動応答を行うのは以下のいずれかの条件を満たす場合のみです。その他のステート、及び `ConfigurationValue` が下記と異なる `Set Configuration` リクエストに対しては自動応答を行いません。

(1) デバイスステートがアドレスステートで、`Set Configuration` リクエストを受信した。

(2) デバイスステートが構成ステートで、`Set Configuration` リクエストを受信した。

ノーデータコントロール転送処理例を、図 4-10に示します。

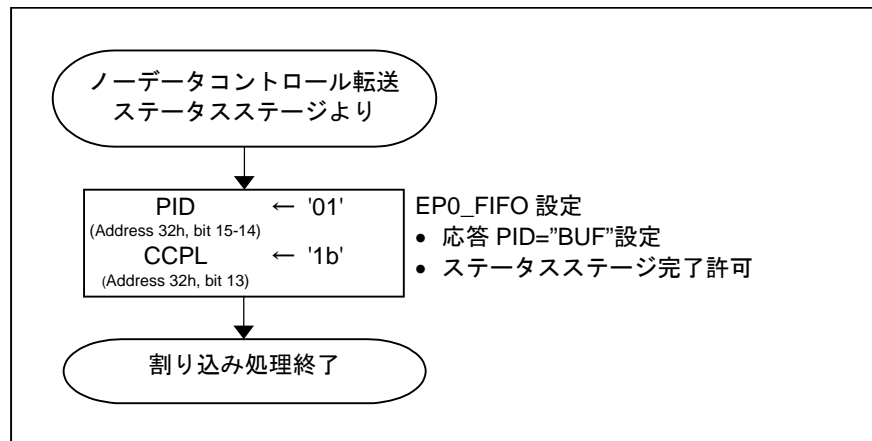


図 4-10 ノーデータコントロール転送処理例

5. データ転送

5.1. データ転送概要

データ転送は片方向転送で、少なくとも2つのパケット(トークンパケット、データパケット)により構成されます。ホストとデバイス間のデータ転送をデータパケット再送機能により保証する転送では、データパケットの後ろにハンドシェイクパケットが加わり、3パケット構成となります。転送種別は、ホストとデバイス間の通信エラーを検出せず、転送レートを保証する2パケット構成のアイソクロナス転送と、通信エラー検出行うが転送レートを保証されない3パケット構成のバルク転送、そして、通信エラー検出行い、転送レートを保証する3パケット構成のインタラプト転送の3種類があります。

データ転送タイプによる特徴は以下のとおりです。

表 6-1 データ転送タイプ一覧

転送タイプ	バルク転送	インタラプト転送	アイソクロナス転送
実時間性	低い	リアルタイム	高い
転送間隔	非保証	保証	保証
データ転送量	大量	少量	少量-大量
パケット数	3	3	2
Data パケット	Data0/Data1	Data0/Data1	Data0
Handshake	有り	有り	無し
最大パケットサイズ	8、16、32、64	0-64	0-1023
M66291 の機能特性			
連続送信機能 ¹	有効	使用禁止	使用禁止
ダブルバッファ機能	有効	有効	有効
zero-length パケット自動送信機能 ²	有効(連続転送時)	使用禁止	使用禁止
DMA アクセス	有効	有効	有効

データ転送のトランザクションは次のように規定されています。

(1) バルク IN、インタラプト IN :

INトークン、データ(DATA0/1)及びハンドシェイクパケットにより構成されます。

IN トークン → DATA(0/1) → ACK

(2) アイソクロナス IN :

INトークン及びデータ(DATA0)により構成されます。

IN トークン → DATA(0)

(3) バルク OUT、インタラプト OUT :

OUTトークン、データ(DATA0/1)及びハンドシェイクパケットにより構成されます。

OUT トークン → DATA(0/1) → ACK

(4) アイソクロナス OUT :

OUTトークン及びデータ(DATA0)により構成されます。

OUT トークン → DATA(0)

これら一連のトランザクションを IN(DATA0/1)、OUT(DATA0/1)と表記します。

¹ インタラプト転送、アイソクロナス転送設定時は、EPI_RWMD[連続送受信モード]=0'に設定してください。

² (バルク OUT 転送かつ EPI_RWMD=1')の時以外は、EPI_NULMD[zero-length パケット自動送信モード]=0'に設定してください。

各データ転送は次のようなトランザクション構成で通信が行われます。

- (1) バルク IN 転送、インタラプト IN 転送 :

IN(DATA0) IN(DATA1) IN(DATA0) IN(DATA1) ... IN(DATA0/1)

- (2) アイソクロナス IN 転送 :

IN(DATA0) IN(DATA0) IN(DATA0) IN(DATA0) ... IN(DATA0)

- (3) バルク OUT 転送、インタラプト OUT 転送 :

OUT(DATA0) OUT(DATA1) OUT(DATA0) OUT(DATA1) ... OUT(DATA0/1)

- (4) アイソクロナス OUT 転送 :

OUT(DATA0) OUT(DATA0) OUT(DATA0) OUT(DATA0) ... OUT(DATA0)

5.2. M66291 によるデータ転送の特徴

M66291 は、USB 通信で規定されている全タイプのデータ転送が可能です。また、各エンドポイントにて行うデータ転送プログラムが簡単に記述できるように以下の機能を備えています。

- (1) 豊富なエンドポイント割り込み 【本資料2.5、5.3参照】
- (2) 合計 3K バイトの FIFO バッファ 【本資料2.6、5.4、5.5参照】
- (3) 連続送受信機能(バルク転送) 【本資料2.6参照】
- (4) エンドポイント毎に転送条件が設定可能 【本資料2.6参照】

M66291 はダブルバッファ機能を備えています。ダブルバッファを使用する設定を行えば、デバイスがホストとデータ通信中でも、FIFO バッファ空きを待つことなくバッファアクセスを行うことが可能です。ダブルバッファ機能を使用することにより USB バスのトラフィックを向上させることができます。

また、M66291 はバルク転送に対し連続送受信機能を使用することができます。データ転送における連続送受信機能は、最大 1K バイトまで割り込みを発生せず連続してデータを送受信する機能です。連続送受信機能を用いることで、バルク転送では 1 トランザクションごとに FIFO バッファアクセスを行う必要がなくなり、CPU 稼働率を高めることができます。

M66291 は、FIFO バッファがフルでホストからのデータを受信できない状態、または FIFO バッファが空でホストからの IN トークンに対してデータ送信を行えない状態のときに、ホストからのトークンパケットを受信すると自動的に”NAK”応答を行います。

エンドポイント初期設定制御手順例を図 5-1 に示します。

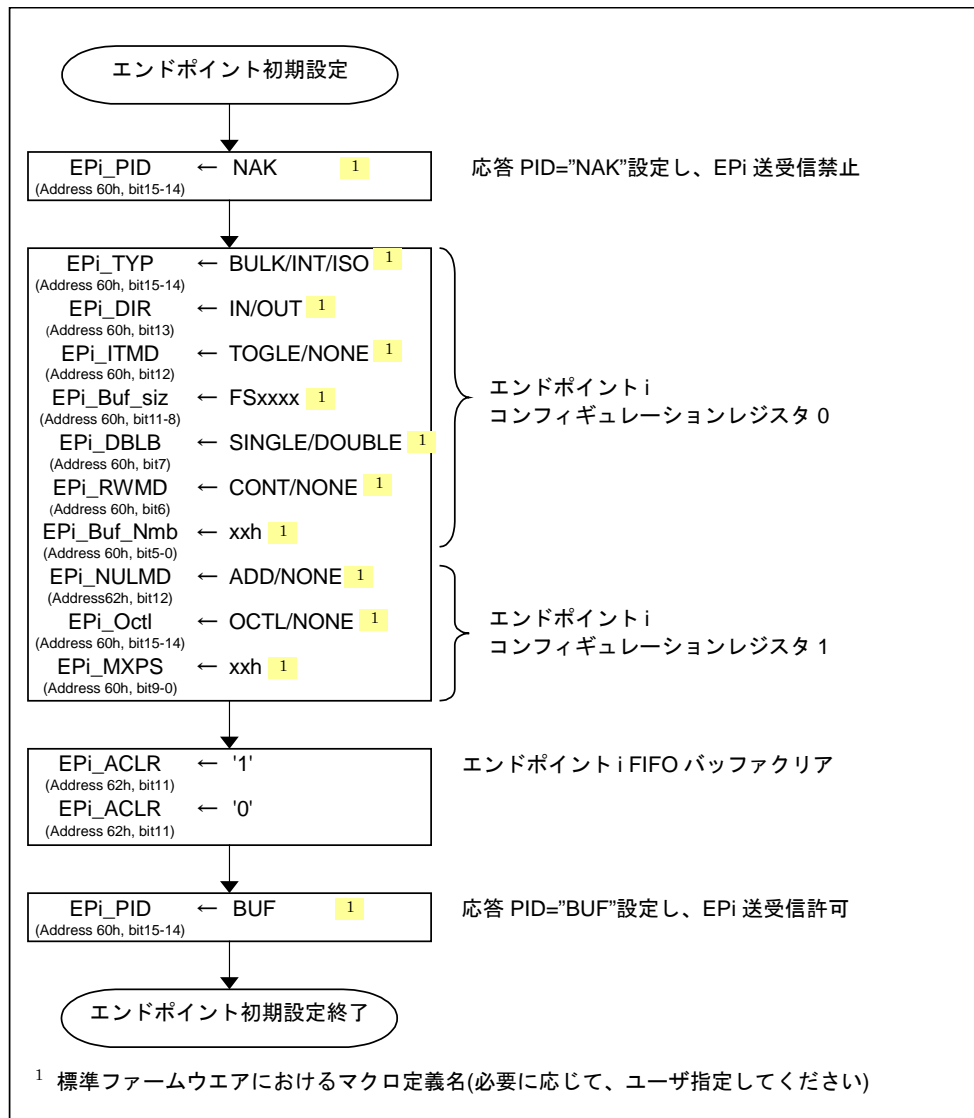


図 5-1 エンドポイント初期設定制御手順例

5.3. データ送受信割り込み

M66291 はバッファレディ、バッファノットレディ、バッファエンプティの 3 種類のデータ転送割り込み機能を備えています。また、各割り込みはステータスレジスタ 1-3 により、割り込み要因の発生したエンドポイントを認識することが可能です。データ転送割り込みは転送タイプ、転送方向及びアクセス方法により発生要因が異なりますのでご注意ください。要因の差異に関しては [2.5 データ転送割り込み](#) を参照ください。

データ送受信制御手順例を図 5-2 に示します。

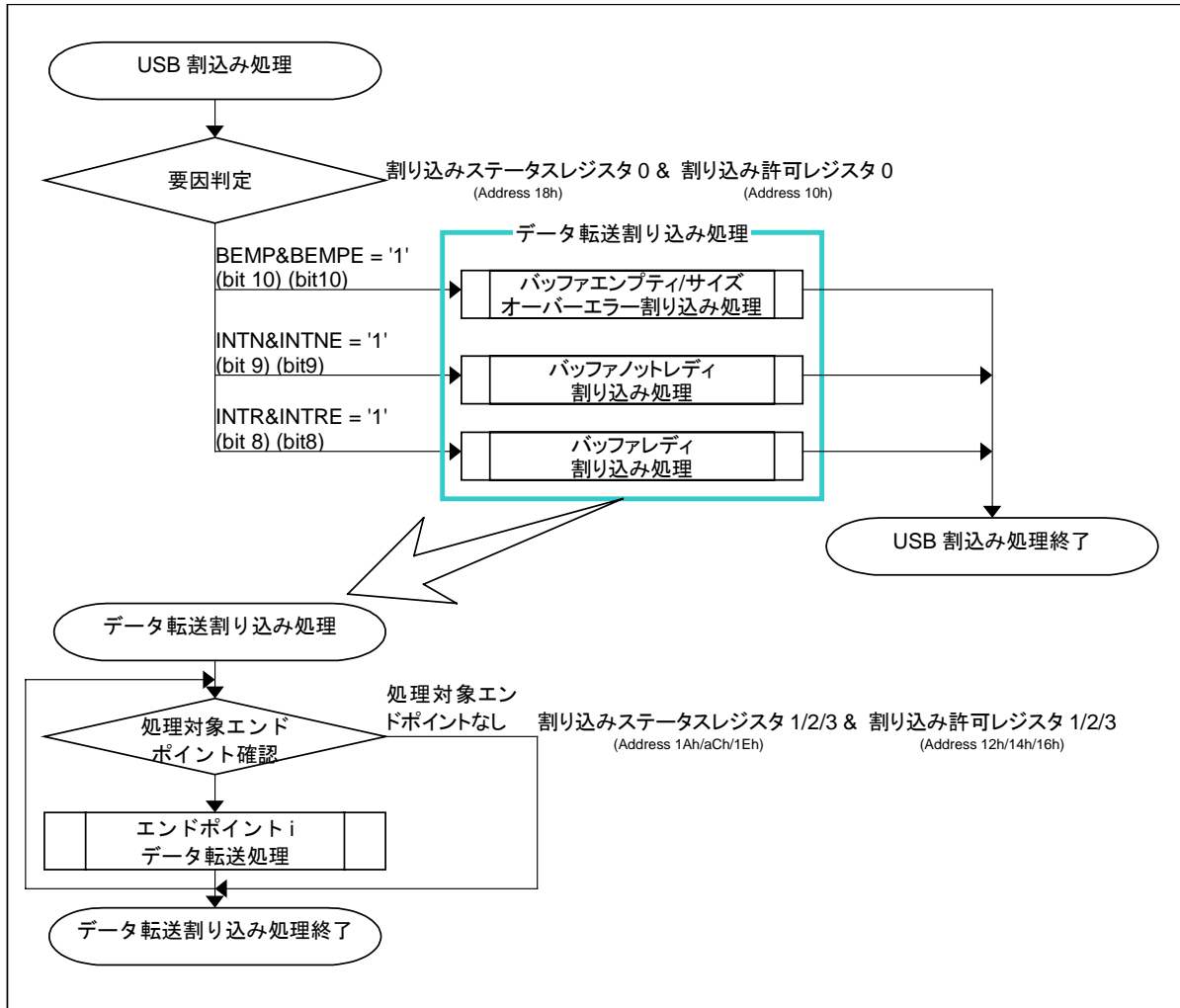


図 5-2 データ送受信制御手順例

5.4. FIFO アクセス(データ送信:IN 方向)

デバイス側に、ホストへの送信データが存在する場合は、IN トランザクションによりデータを送信します。USB 通信はホスト PC のサンプリングによりデータ転送が行われるため、デバイス側アプリケーションは送信データが発生したら M66291 の FIFO バッファに送信するデータを書き込み、ホストからの IN トークンを待ちます。データ送信が終了し、FIFO バッファに CPU 側からの書き込みが可能となった時点で、M66291 は CPU に対してバッファレディ割り込みを通知します。要因の詳細に関しては [2.5 データ転送割り込み](#) を参照ください。

データ送信設定制御手順例(CPU アクセス)を図 5-3に示します。

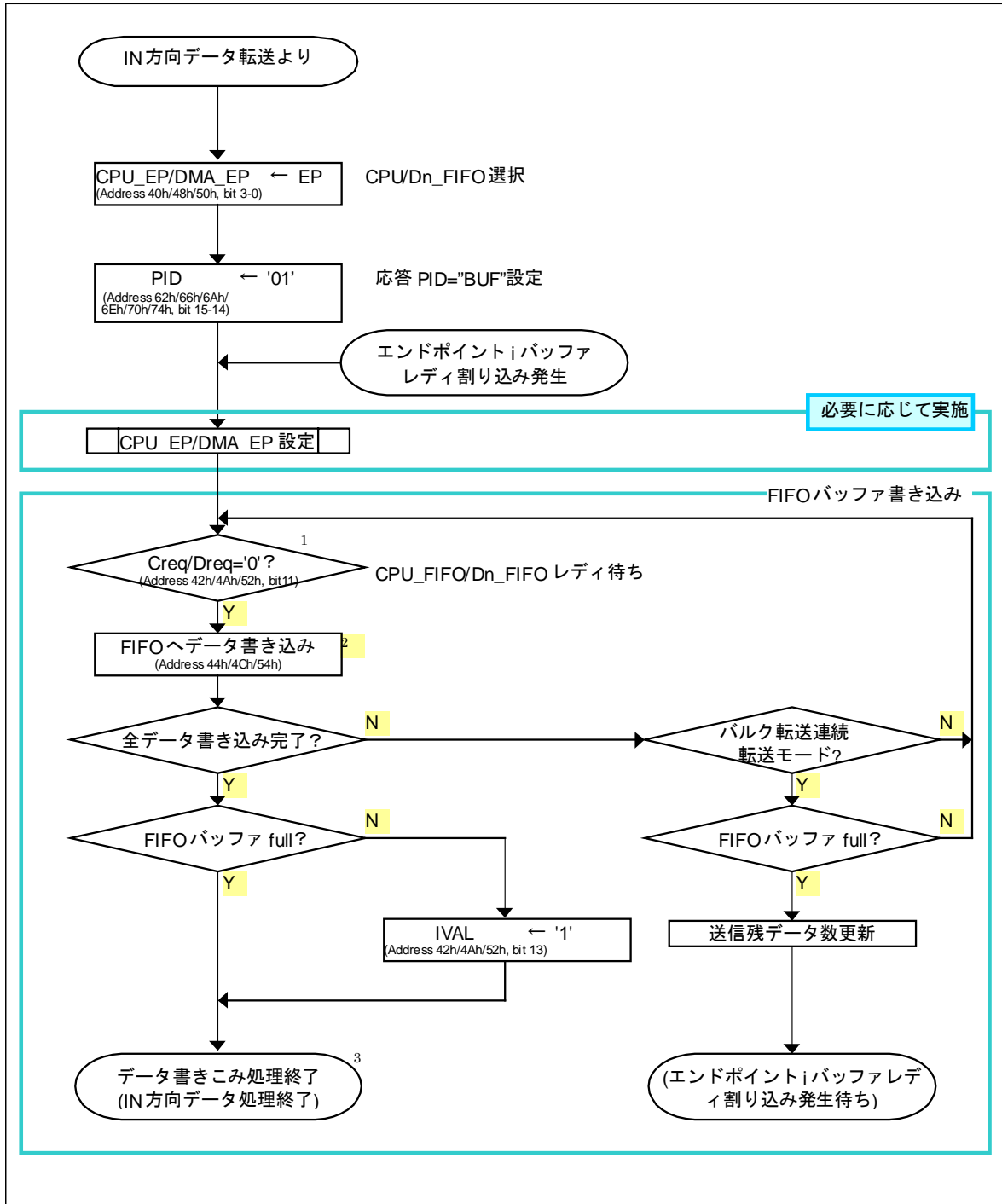


図 5-3 データ送信設定制御手順例(CPU アクセス)

¹ FIFO データレジスタへのアクセスサイクルがデータシートを満たす場合は、この確認は省略可能です。

² 16bit バス幅接続のシステムで、奇数バイト数のデータ書き込みを行う場合は、16ビットモードで最後の1バイト

直前まで書き込みし、**Octl='1'**設定により 16bit→8bit に切り替えて最後の 1 バイトの書き込みを行います。16 ビットモード時にバイトスワップ機能(**BSWP='1'**)を使用する場合は、**BSWP='0'**に戻してから **Octl='1'**設定を行ってください。

- 3 バルク転送連続送信モードの場合、最後の 1 パケットがショートパケットではない場合に、アプリケーションによってはこの後の**エンドポイント i バッファレディ**割り込みで **zero-length** パケット送信準備が必要です。

5.5. FIFO アクセス(データ受信:OUT 方向)

デバイスがホスト PC からデータを受信する場合は、OUT トランザクションによりデータ転送を行います。USB 通信はホスト PC のサンプリングによりデータ転送が行われるため、デバイス側アプリケーションでは、あらかじめ M66291 の FIFO バッファを空き状態(受信可能状態)にしておき、ホストからの OUT トークンを待つ処理を行ってください。M66291 は、受信完了により FIFO バッファの読み出しが可能となったことを、CPU に対してバッファレディ割り込み発生させることによって通知します。(要因の詳細に関しては [2.5 データ転送割り込み](#) を参照)

データ受信設定制御手順例を図 5-4に示します。

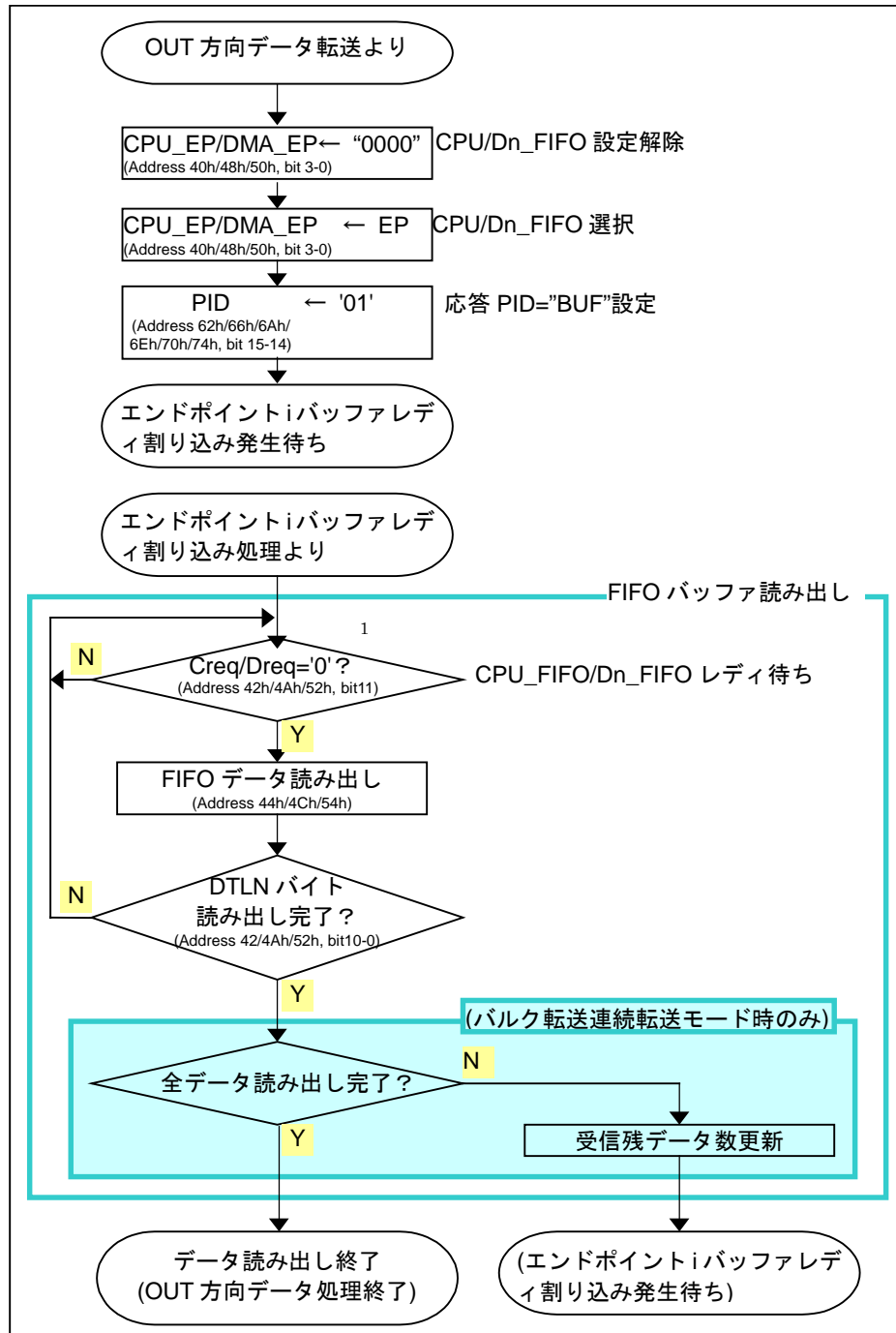


図 5-4 データ受信設定制御手順例

¹ FIFO データレジスタへのアクセスサイクルがデータシートを満たす場合は、この確認は省略可能です。

6. USB 特殊処理

6.1. バスリセット信号処理

6.1.1. USB Specification 概要

デバイスは、USB バス上の USB バスリセット信号(2.5 μ s 以上の SE0)を検出後、すでに割り振られたアドレスを Default アドレスに変更するとともに、Default ステートにステート遷移する必要があります。

6.1.2. M66291 の機能

M66291 は、USB バスリセット信号を検出すると、Default ステートに遷移し、デバイスステート遷移割り込み(DVSQ='001')を発生させます。このとき、M66291 は USB_Address レジスタ(Address 08h)を自動的に Default アドレスに書き換え、M66291 レジスタの一部を初期化します。

USB バスリセット検出に伴う M66291 レジスタ初期化については、M66291 データシート【レジスタ一覧】を参照ください。

6.1.3. デバイス側アプリケーションの動作

デバイス側アプリケーションは、USB バスリセット検出(デバイスステート遷移割り込み(DVSQ='001')検出)後、ホストからのリクエスト(バスエニュメレーション)に応答する準備を行う必要があります。

USB バスリセット信号検出処理例を図 6-1に示します。

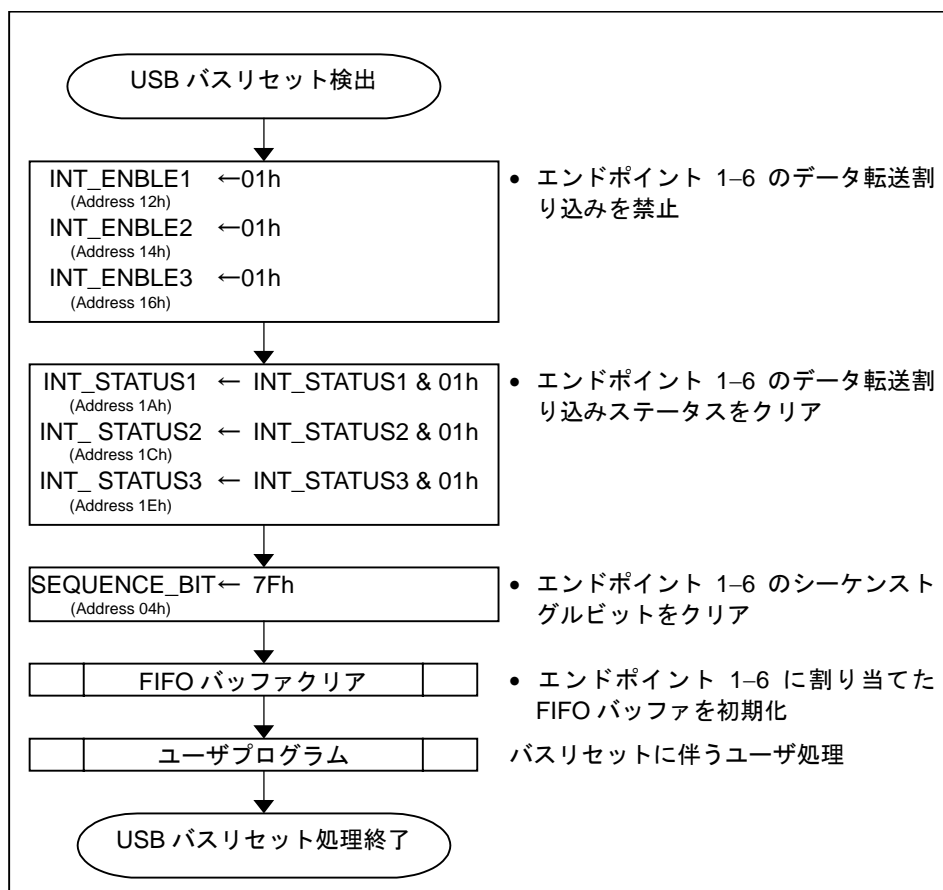


図 6-1 USB バスリセット信号検出処理例

6.2. サスペンド信号処理

6.2.1. USB Specification 概要

USB バス上でサスペンド信号(3.0ms 以上のアイドル状態)を検出すると、デバイスは USB specification で規定されている任意のステートから Suspend ステートへ遷移を行う必要があります。また、Suspend ステートにあるデバイスは、通常 USB バスから 500 μ A 未満の電流しか引き出すことができません。

デバイスが Suspend ステートから復帰するためには、何らかのバス アクティビティが発生するか、リモートウェイクアップ機能を使用してホストに要求を行う必要があります。

6.2.2. M66291 の機能

M66291 は、サスペンド信号を検出すると、Suspend ステートに遷移し、デバイスステート遷移割り込み(DVSQ='1xx')を発生させます。

M66291 自身の消費電力を小さくするために、以下の設定が有効です。具体的な消費電力値については、M66291 データシート【電気的特性】を参照ください。

- (1) 内部クロック供給禁止(SCKE='0')¹
- (2) 発振バッファ停止(XCKE='0')¹
- (3) USB トランシーバ動作禁止(USBPC='0')¹

M66291 は内部クロック供給停止中、発振バッファ停止中、又は、USB トランシーバ動作禁止中にも、RESM 割り込みによるバス アクティビティを検出することが可能です。(6.3 レジューム (バス アクティビティ検出)、6.4 Suspend ステート中の USB バスリセット検出を参照)

6.2.3. デバイス側アプリケーションの動作

デバイスが Bus powered を使用している場合、デバイス側アプリケーションは、サスペンド信号検出(Suspend ステートへのデバイスステート遷移割り込み(DVSQ='1xx')検出)後、低消費電力モードに移行する必要があります。低消費電力モードに移行する場合、レジューム検出時にデバイス動作を通常に戻す必要があるため、レジューム検出が可能な設定(RESM 割り込みの許可)を行ってください。

また、デバイスがリモートウェイクアップ機能をサポートしている場合は、システム仕様に従い、リモートウェイクアップ信号送出機能を実装する必要があります。6.5 リモートウェイクアップ処理も参照ください。

サスペンド検出処理例を図 6-2 に示します。

¹ USB 動作許可レジスタ(Address 00h)

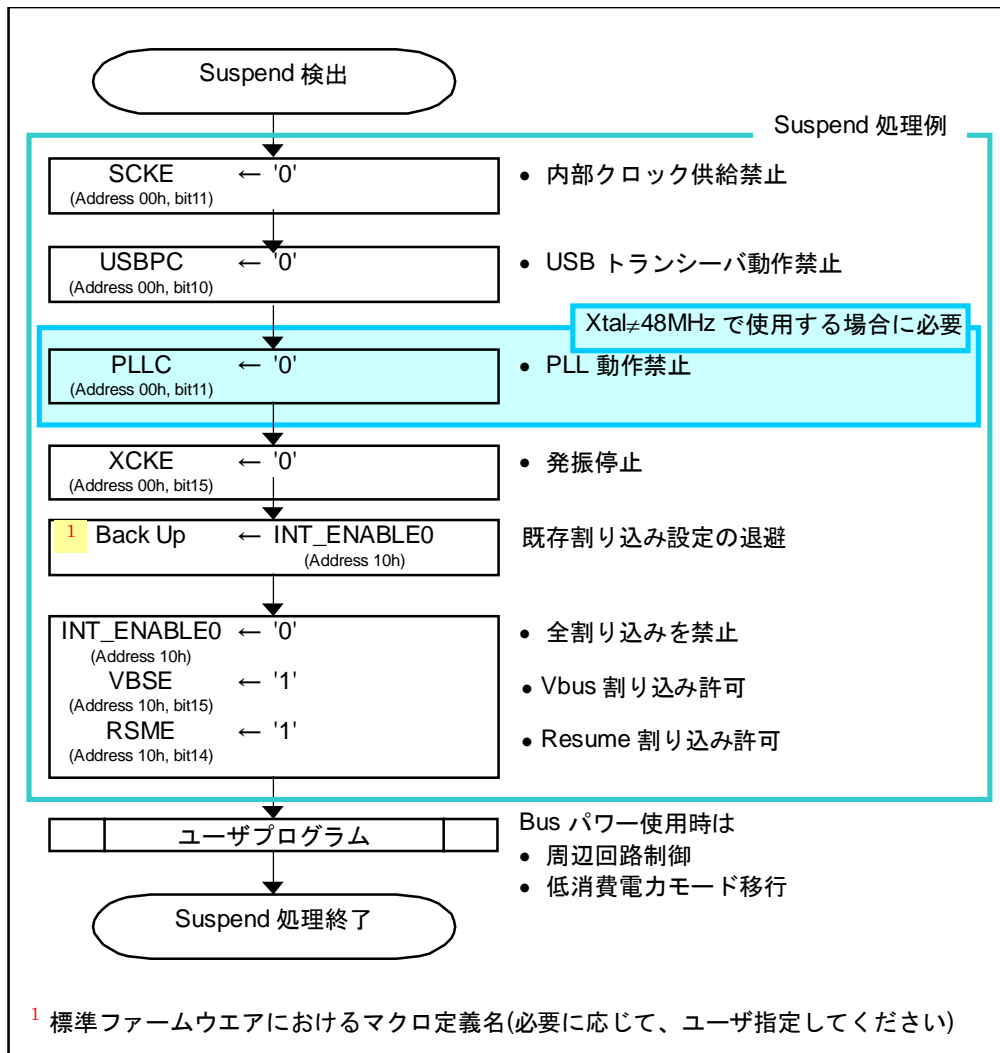


図 6-2 サスペンド検出処理例

6.3. レジューム (バス アクティビティ検出)処理

6.3.1. USB Specification 概要

USB バスが Suspend 状態の時に USB バスの状態変化があった場合に、デバイスはレジュームを検出する必要があります。

6.3.2. M66291 の機能

デバイスステートが Suspend ステートである時に USB バスの状態変化(“J”→“K”もしくは“SE0”)を検出した場合に、M66291 はレジューム検出割り込み(RESM)を発生させ、サスペンド検出直前のデバイスステートに自動的に遷移します。

レジューム検出割り込みは USB バス状態の変化を検出しますので、M66291 の内部クロック供給停止中、発振バッファ停止中、又は、USB トランシーバ動作禁止中にも、RESM 割り込みによるバス アクティビティ検出が可能です。

6.3.3. デバイス側アプリケーションの動作

サスペンド時に低消費電力モードに移行する等の処理を行うシステムの場合、レジューム検出時にデバイス動作を通常に戻す必要があります。

また、レジューム検出割り込みのステータスフラグのクリア方法は、内部クロック供給時と停止時で異なりますのでご注意ください。

- (1) 内部クロック供給時: RESM='0'¹を書き込んでください
- (2) 内部クロック停止時: RESM='0'、RESM='1'を連続して書き込んでください

レジューム検出処理例を図 6-3に示します。

¹ 割り込みステータスレジスタ 0(Address 18h)、bit14

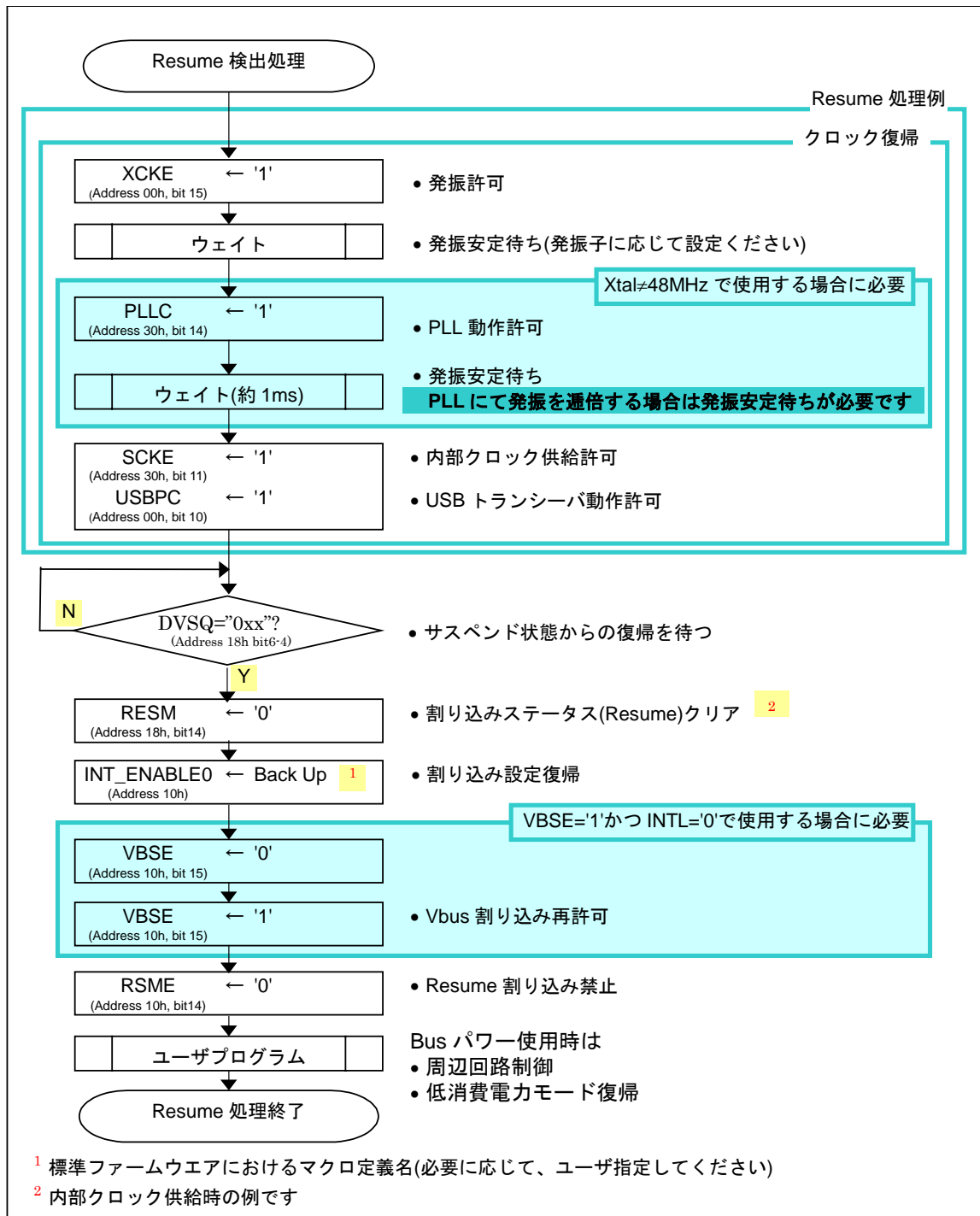


図 6-3 レジューム検出処理例

6.4. Suspend ステート中の USB バスリセット検出

6.4.1. USB Specification 概要

USB バスリセット信号もバス アクティビティの一種ですので、この場合、デバイスは、サスペンド状態からのレジューム、及び、USB バスリセット信号の検出を行う必要があります。

6.4.2. M66291 の機能

M66291 の内部クロック供給を停止している場合は、レジューム検出は行いますが、USB バスリセット信号の検出を行うことができません。したがって、Suspend ステート中に USB バスリセットが発生した場合は、以下のようなシーケンスとなります。

- (1) レジューム検出
- (2) FW によるレジューム処理(M66291 の内部クロック供給、USB トランシーバ動作等)
- (3) USB バスリセット検出(デバイスステート(DVSSQ='001': Default)遷移割り込みを発生)

6.4.3. デバイス側アプリケーションの動作

デバイス側アプリケーションは、ホストからの USB リセット信号発行中に、レジューム処理を終わらせる必要があります。USB Specification によれば、ホスト PC が発行する USB バスリセット信号 10ms 以上に対し、デバイス側(M66291)が USB バスリセットを検出するために 2.5 μ s 必要ですので、レジューム検出後遅くても 9.975ms 以内に内部クロック復帰処理が終了するように処理タイミングを設計してください。

Suspend ステート中の USB バスリセット検出処理例を図 6-4に示します。

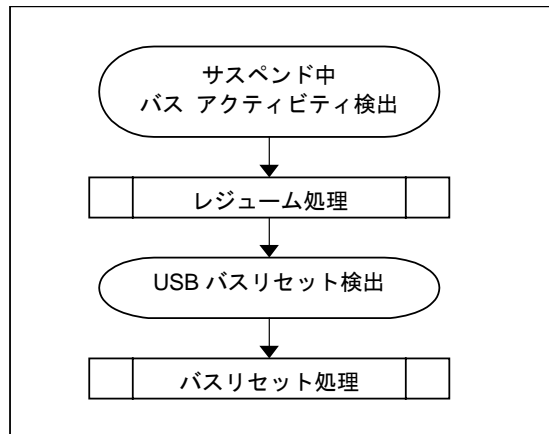


図 6-4 Suspend ステート中の USB バスリセット検出処理例

6.5. リモートウェイクアップ処理

6.5.1. USB Specification 概要

デバイスステートが Suspend ステートであるときに、デバイス側の要因により USB 通信を再開したい場合は、デバイスからリモートウェイクアップ信号を送出させホストに対してレジューム要求を促すことができます。

リモートウェイクアップ信号は、USB バスがアイドル状態に遷移した後、アイドル状態を 5ms のあいだ保持した後に出力する必要があります。また、リモートウェイクアップレジューム信号の出力期間は 1ms 以上 15ms 以内と定められています。

リモートウェイクアップ機能には使用条件があります。以下(1)、(2)を満たす場合のみ、デバイスはリモートウェイクアップを行うことができます。

- (1) 構成ステートから遷移した Suspend ステートでの使用であること
- (2) 当該デバイスに対し、SetFeature リクエストによって、リモートウェイクアップが許可されていること

6.5.2. M66291 の機能

M66291 のリモートウェイクアップ機能は、リモートウェイクアップ出力設定後、2ms 間のアイドルステートを保った後、10ms 間の K ステート出力を行います。Suspend 信号検出時点で USB バスがアイドル状態に遷移した後 3ms 経過しているため、上記 2ms 間のアイドル状態保持機能により、デバイス側アプリケーションが Suspend 検出直後にリモートウェイクアップ信号の出力設定を行ったとしても、バスアイドル 5ms 保持の規格を遵守することが可能です。

6.5.3. デバイス側アプリケーションの動作

デバイス要因により Suspend ステートからのレジュームを行いたい場合は、レジューム要因発生後レジューム検出割り込み発生時と同様の処理を行い、その後、リモートウェイクアップ出力処理を行います。

リモートウェイクアップ処理例を図 6-5に示します。

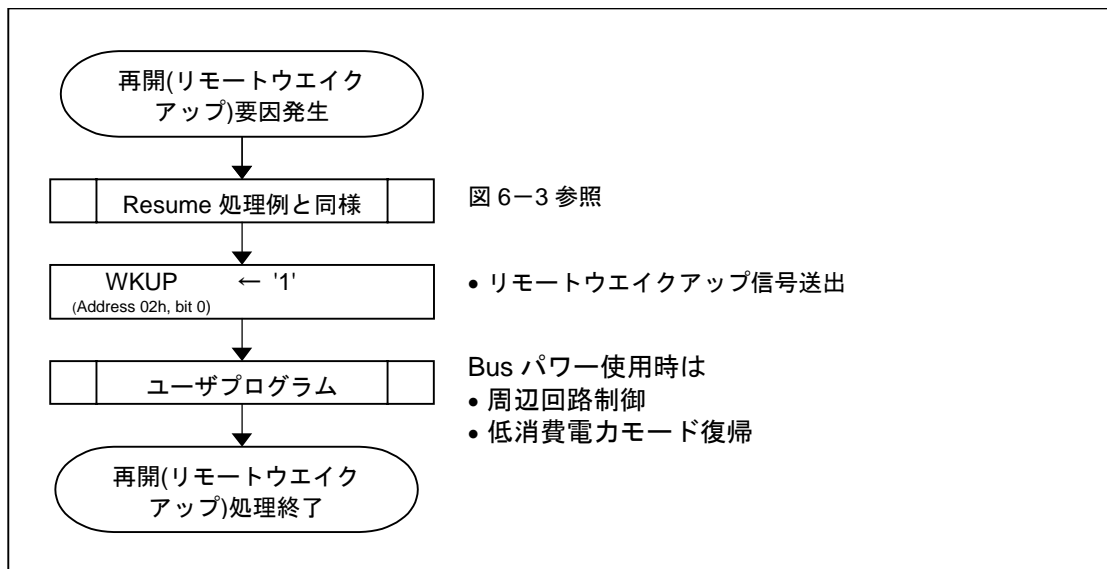


図 6-5 リモートウェイクアップ処理例

6.6. Vbus 変化の検出処理

6.6.1. USB Specification 概要

full speed デバイスは、VBUS 信号が 4.01V 以上になった後 100ms 以内に D+信号を pull up しなければなりません。ホストは、full speed デバイスの D+信号が pull up された後、少なくとも 100ms 待つてから(D+信号の状態が安定するのを待つため)、USB 通信を開始します。

デバイスは、VBUS が供給されていない時には、D+、D-信号を pull up してはいけません。

6.6.2. M66291 の機能

M66291 は、Vbus 入力端子の状態変化を検出(“H”→”L”、”L”→”H”両エッジを検出可能)した場合に Vbus 割り込みを発生(VBUS=’1’)させます。立ち上がり/立下りのどちらのエッジを検出したかは、Vbus 入力ビット(Vbus)で確認することが可能です。M66291 の内部クロック供給が停止されていても、M66291 は VBUS 割り込みを発生させることができ、Vbus 端子の状態を Vbus 入力ビットに反映させることができます。

M66291 の TrON 端子の出力は Vbus 端子の状態が反映されます。Vbus 端子に USB コネクタの VBUS 信号を接続し、かつ、TrON 端子を使用して D+信号の pull up を行うデバイスの場合、self powered デバイスであっても、USB ケーブルが抜かれた時に D+信号の pull up をやめるための特別な回路、FW 処理は不要です。

6.6.3. デバイス側アプリケーションの動作

USB ケーブルが抜かれた場合に何らかの処理が必要なシステム(self powered で Vbus 供給がない場合に消費電力を落とす必要がある、又は、bus powered で電源 off の前に処理を行う必要がある等のシステム)では、USB ケーブルの挿抜を Vbus 入力端子の状態変化で検出し、処理を行う必要があります。

Vbus 検出処理例を図 6-6に示します。省電力処理/復帰処理は消費電力を小さくする場合の例です。

Vbus 割り込みを使用する場合は、M66291 の初期処理の中で、Vbus 割り込みを許可(VBSE=’1’)する必要があります。なお、Vbus 入力ビット(Vbus)はチャタリング除去のうえ状態確認を行う必要があります。チャタリング除去処理の中で、VBUS(Vbus 割り込みステータスビット)のクリアを行ってください。

また、VBUS 割り込みのステータスフラグのクリア方法は、内部クロック供給時と停止時で異なりますのでご注意ください。

- (1) 内部クロック供給時: VBUS=’0’¹を書き込んでください
- (2) 内部クロック停止時: VBUS=’0’、VBUS=’1’を連続して書き込んでください

VBUS クリア処理例を図 6-7に示します。

¹ 割り込みステータスレジスタ 0(Address 18h)、bit15

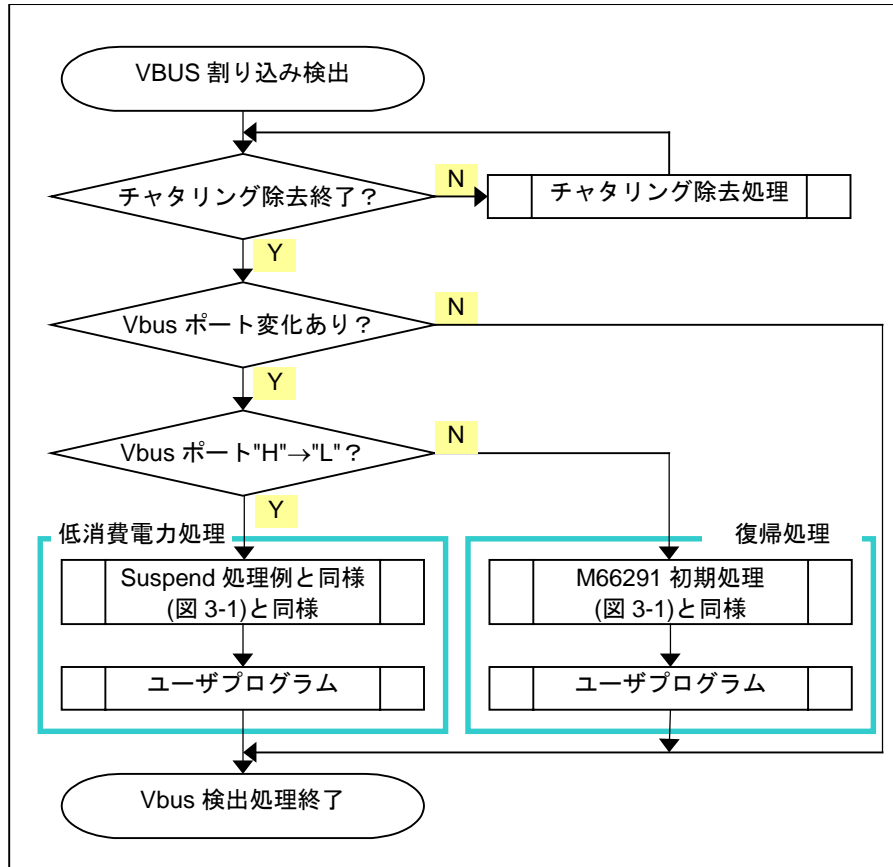


図 6-6 Vbus 検出処理例

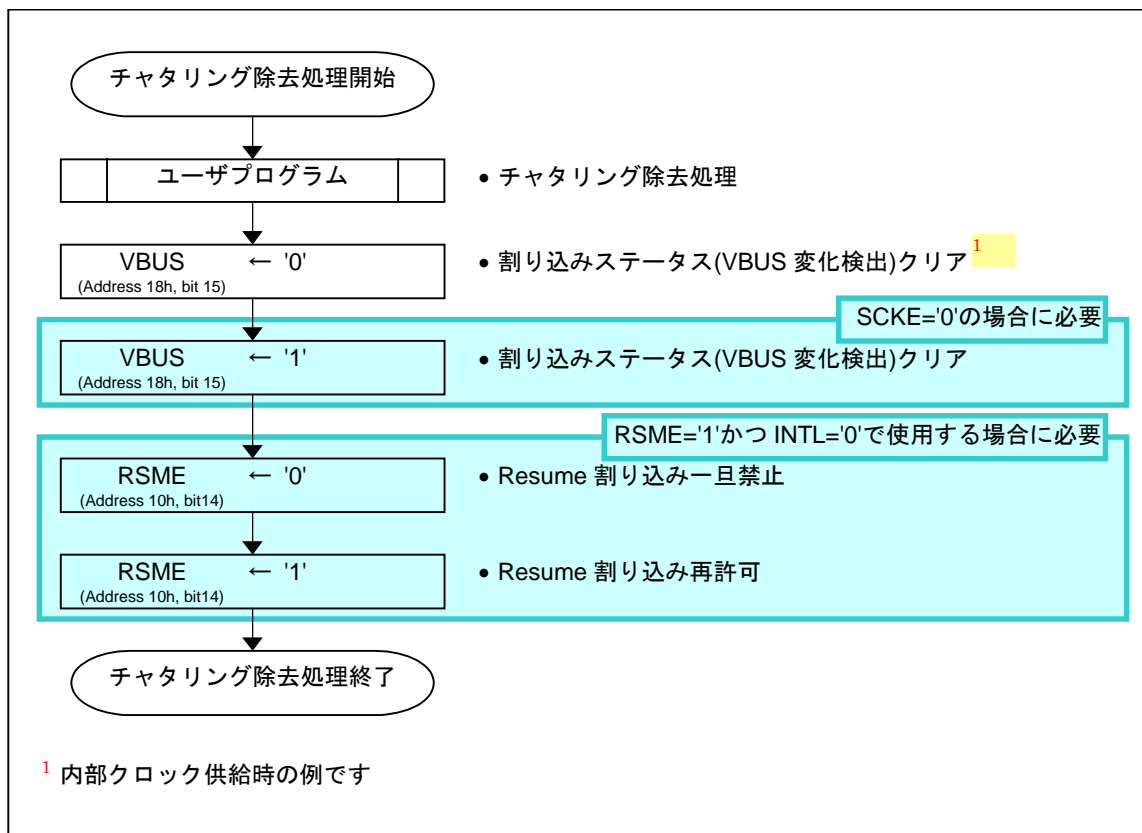


図 6-7 VBUS クリア処理例

7. 応用例

7.1. CPU との接続方法(16bit バス幅)

M66291 が接続される CPU からのアクセスについて考える場合、CPU のエンディアンが問題になります。M66291 に接続が想定される CPU のエンディアンを表 7-1 に列挙します。

表 7-1 M66291 に接続が想定される CPU とそのエンディアン

No.	ビット エンディアン	バイト エンディアン	CPU 例
1	リトル	リトル	7700 シリーズ、7900 シリーズ、M16C ファミリー SH ファミリーなど
2	リトル	ビッグ	H8 シリーズ、他社マイコン
3	ビック	リトル	(一般的ではないので掲載省略)
4	ビック	ビック	M32R ファミリー

この章では、16bit バス幅での各 CPU と M66291 との結線方法、及びアクセス方法について説明します。M66291 へのアクセスは、レジスタへのアクセス、FIFO バッファへのアクセスに大別され、FIFO バッファへのアクセスには、CPU によるアクセスと DMA 転送によるアクセスがあります。また、奇数バイト数のデータを FIFO バッファに書きこむ場合の 1 バイトアクセスについての考察も述べます(奇数バイト数データの読み出しについては、[2.6 FIFO 制御](#)、[2.7.2 16bit バス幅接続時に Dn_FIFO データレジスタで奇数バイト数データの読み出し/書き込みを行う方法](#)を参照ください)。

各種エンディアンの CPU と M66291 の接続性を表 7-2 にまとめます。

表 7-2 PU のエンディアンによる M66291 の接続性(16bit 接続時)

エンディアン		接続方法	BSWP 設定値 ²	アクセス ¹	
ビット	バイト			レジスタ	FIFO
リトル	リトル	7.1.1章の方法	'0'	ok	ok
リトル	ビッグ	7.1.2.1章の方法	'1'	ok	ok
		7.1.2.2章の方法	'0'	invert	ok
ビッグ	ビッグ	7.1.3.1章の方法	'1'	ok	ok
		7.1.3.2章の方法	'0'	invert	ok

M66291 と CPU を結線するときに、データのバイト単位での逆転がなく、ありのままで転送できるかどうかは大きな問題です。M66291 では、プログラムを工夫することによりこの問題を解決することが可能です。

実際に設計されるときは、お客様の使い方にあわせて結線方法を選択する必要があります。

¹ ok:データのありのまま読み出せる、或いは書き込めることを示します。

invert:バイトが逆転して読み出せる、或いは書き込めることを示します。

² 16bit アクセス時の設定値を示します(8bit アクセス時は必ず BSWP='0'に設定してください)。

7.1.1. ビットエンディアンがリトル、バイトエンディアンがリトルの CPU の場合(16bit バス幅)

ルネサスの 7700 シリーズ、7900 シリーズ、M16C ファミリ、SH ファミリなどはこの種類の CPU です。

<接続>

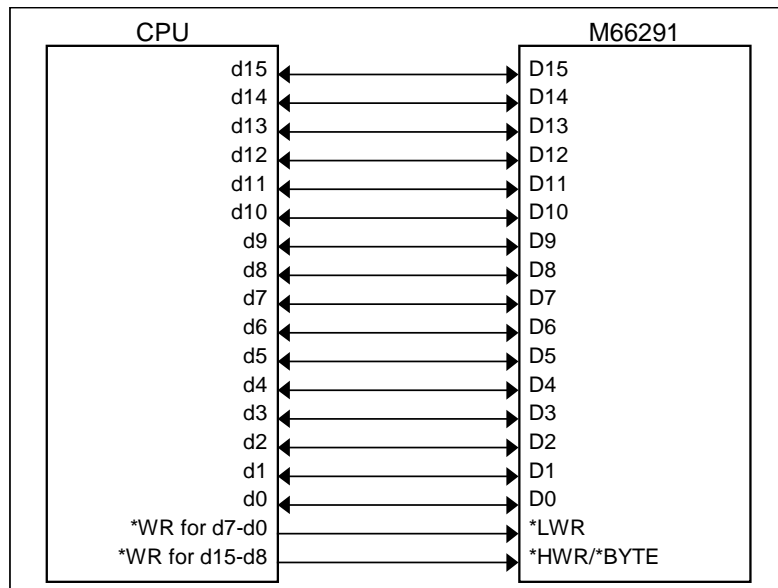


図 7-1 ビットエンディアン:リトル、バイトエンディアン:リトルの接続例(16bit バス幅)

<レジスタアクセス(FIFO_DATA レジスタ以外)>

読み出し:M66291のレジスタのデータがそのまま、CPUのレジスタ或いはメモリに読み出されます。

書き込み:CPUのレジスタ或いはメモリ上のデータがそのまま、M66291のレジスタに書き込まれます。

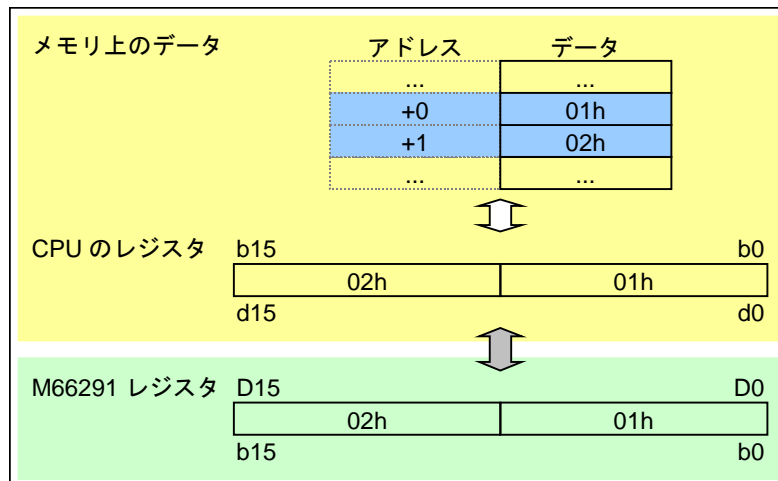


図 7-2 ビットエンディアン:リトル、バイトエンディアン:リトルのレジスタアクセスイメージ(16bit バス幅)

<FIFO バッファ(FIFO_DATA レジスタ)アクセス>

(1) M66291 の FIFO に対する CPU アクセス

基本的に、BSWP(バイトスワップモード;Address 40h, bit 12)='0'、Octl(レジスタ8bitモード;Address 40h, bit 6)¹='0'を設定してアクセスします。

奇数バイト数データの最後の1バイトのみ、BSWP='0'、Octl='1'を設定し、CPUのd7-d0に有効データを、CPUのd15-d8にダミーデータを出力し、16ビット単位で書き込みます。

読み出し:M66291のFIFOレジスタから読み出したデータがそのまま、順番にメモリに格納されます。

書き込み:メモリに格納されたデータがそのまま、順番にM66291のFIFOバッファに書き込まれます。

(2) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。但し、BSWP、Octlは、Dn_FIFO選択レジスタ(Address 48h or 50h, bit 12)上のBSWP、Octl²を使用します。

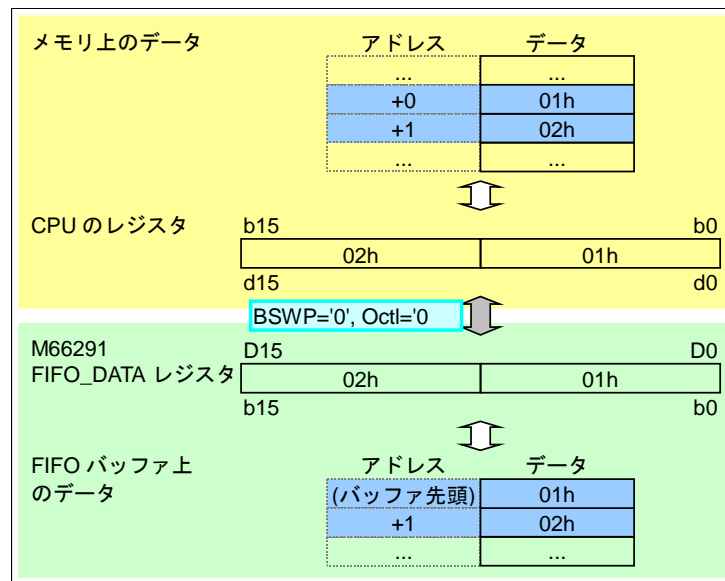


図 7-3 ビットエンディアン:リトル、バイトエンディアン:リトルの FIFO バッファアクセスイメージ 1(16bit バス幅、16 ビットアクセス)

¹ Octl(Address 40h, bit 6)と、Epi_Octl(Address (62+(i-1)*2)h, bit 10)(i は CPU_EP(Address 40h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。どちらか一方が'1'に設定されると、当該エンドポイントの FIFO バッファアクセスは 8 ビットモードになります。

² Octl(Address 48h or 50h, bit 6)と、Epi_Octl(Address (62+(i-1)*2)h, bit 10)(i は DMA_EP(Address 48h or 50h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。

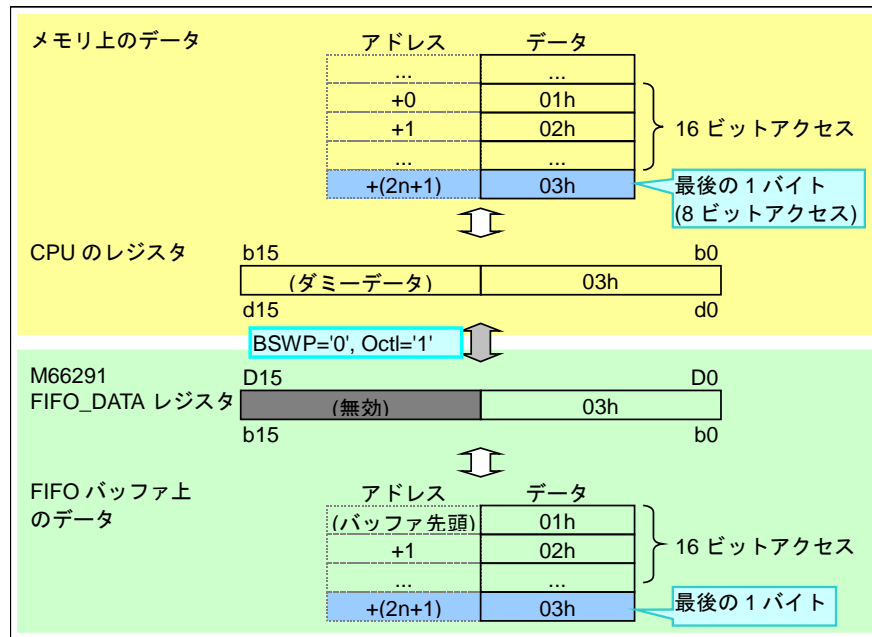


図 7-4 ビットエンディアン:リトル、バイトエンディアン:リトルの FIFO バッファアクセス 2(16bit バス幅、バイト書き込み)

7.1.2. ビットエンディアンがリトル、バイトエンディアンがビッグの場合(16bit バス幅)

ルネサスの H8 シリーズなどはこの種類の CPU があります。M66291 との接続を考える場合には、基本的には方法(1)(7.1.2.1章)を推奨します。参考資料として方法(2)(7.1.2.2章)を併記します。

7.1.2.1. ビットエンディアンがリトル、バイトエンディアンがビッグの場合(16bit バス幅) (1)

<接続>

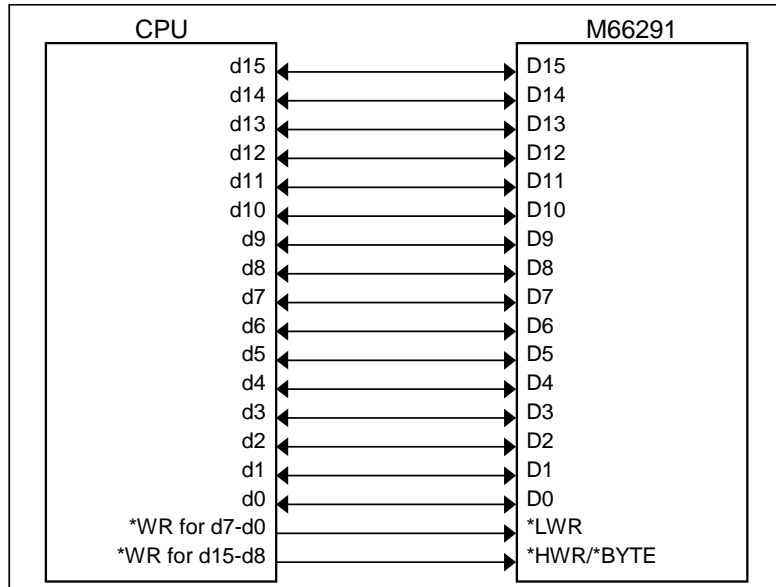


図 7-5 ビットエンディアン:リトル、バイトエンディアン:ビッグの接続図(16bit バス幅) (1)

<レジスタアクセス(FIFO_DATA レジスタ以外)>

読み出し: M66291のレジスタのデータがそのまま、CPUのレジスタ或いはメモリに読み出されます。

書き込み: CPUのレジスタ或いはメモリ上のデータがそのまま、M66291のレジスタに書き込まれます。

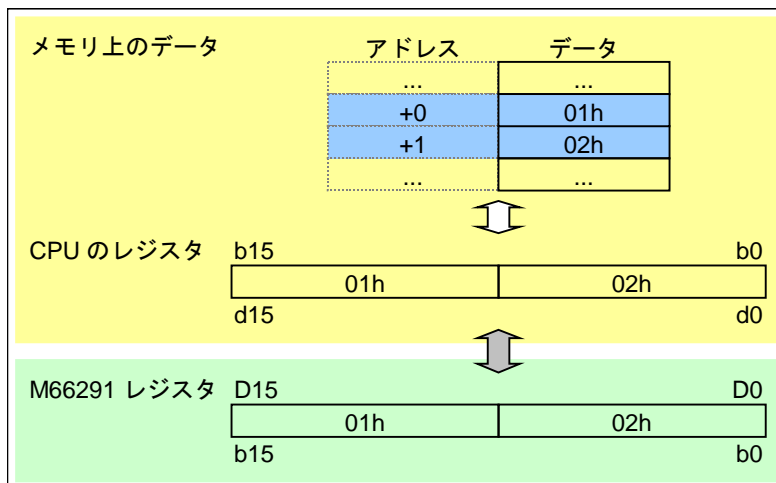


図 7-6 ビットエンディアン:リトル、バイトエンディアン:ビッグのレジスタアクセスイメージ(16bit バス幅) (1)

< FIFO バッファ(FIFO_DATA レジスタ)アクセス >

(1) M66291 の FIFO に対する CPU アクセス

基本的にBSWP(バイトスワップモード;Address 40h, bit 12)=**'1'**、Octl(レジスタ8bitモード;Address 40h, bit 6)¹=**'0'**を設定してアクセスします。BSWP=**'1'**の場合、CPU_FIFO–FIFOバッファ間のデータ転送時にM66291内部でバイト単位で上位・下位を入れ替えます。即ち、読み出し時はFIFOバッファの下位バイトがD15–D8に、上位バイトがD7–0に読み出されます。書き込み時は、D15–D8のデータがFIFOバッファの下位バイトに、D7–D0のデータが上位バイトに書き込まれます。

奇数バイト数データの最後の1バイトのみ、BSWP=**'0'**、かつOctl=**'1'**を設定し、CPUのD7–D0に有効なデータを、CPUのD15–D8にダミーデータを出力し、16ビット単位で書き込みます。

読み出し: M66291のFIFOバッファに格納された順番で、メモリに格納されます。

書き込み: メモリに格納された順番で、M66291のFIFOバッファに格納されます。

(2) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。但し、BSWP、Octlは、Dn_FIFO選択レジスタ(Address 48h or 50h, bit 12)上のBSWP、Octl²を使用します。

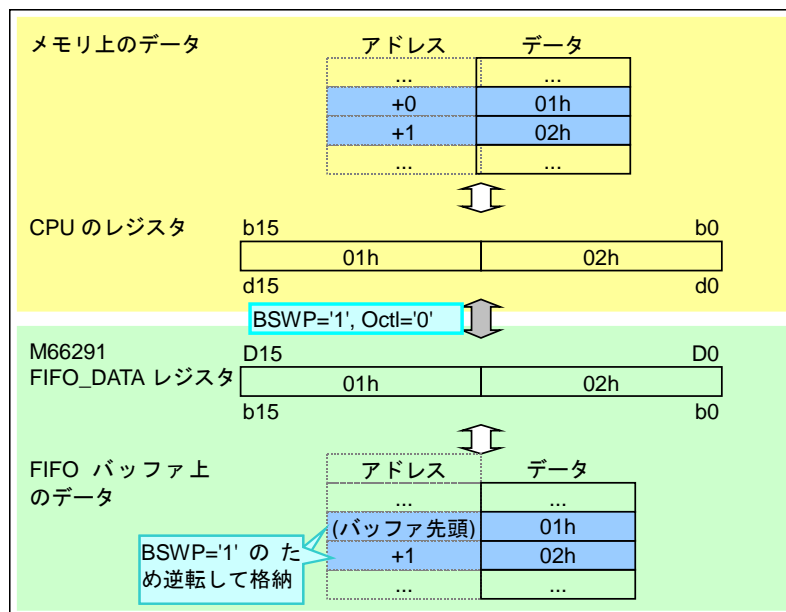


図 7-7 ビットエンディアン:リトル、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(1) (16 ビットアクセス)

¹ Octl(Address 40h, bit 6)と、EPI_Octl(Address (62+(i-1)*2)h, bit 10)(i は CPU_EP(Address 40h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。どちらか一方が'1'に設定されると、当該エンドポイントの FIFO バッファアクセスは 8 ビットモードになります。

² Octl(Address 48h or 50h, bit 6)と、EPI_Octl(Address (62+(i-1)*2)h, bit 10)(i は DMA_EP(Address 48h or 50h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。

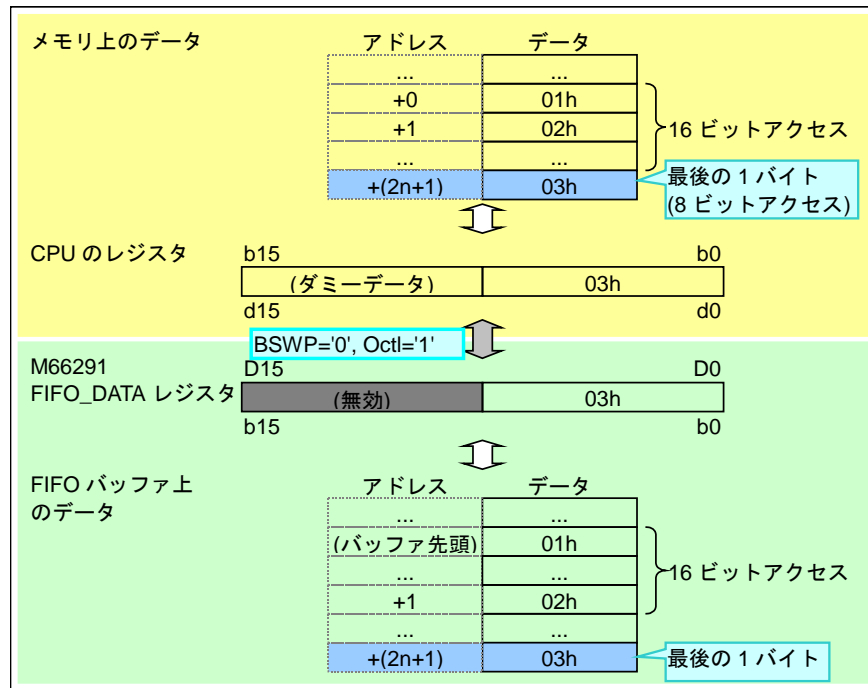


図 7-8 ビットエンディアン:リトル、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(1) (バイト書き込み)

7.1.2.2. ビットエンディアンがリトル、バイトエンディアンがビッグの場合(16bit バス幅) (2)

<接続>

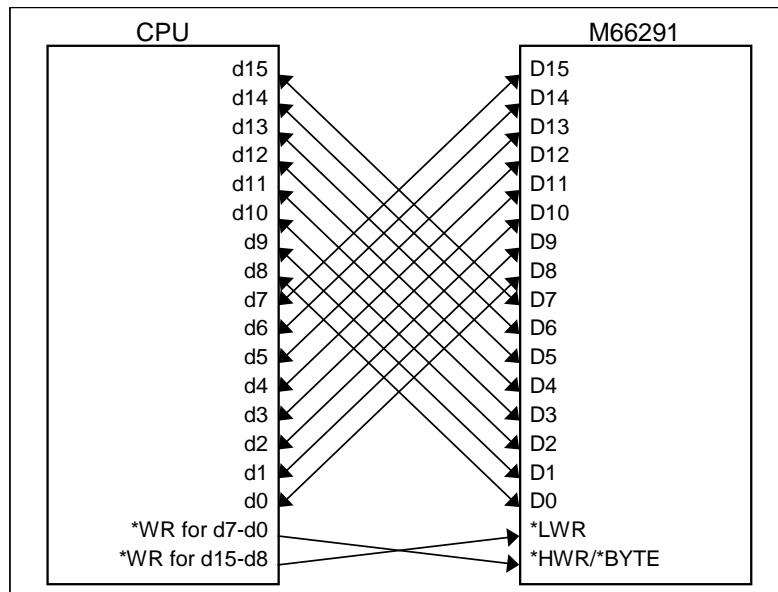


図 7-9 ビットエンディアン:リトル、バイトエンディアン:ビッグの接続図(16bit バス幅) (2)

<レジスタアクセス(FIFO_DATA レジスタ以外)>

読み出し: M66291のレジスタのデータがバイト単位で逆転して、CPUのレジスタ或いはメモリに読み出されます。

書き込み: CPUのレジスタ或いはメモリ上のデータがバイト単位で逆転して、M66291のレジスタに書き込まれます。

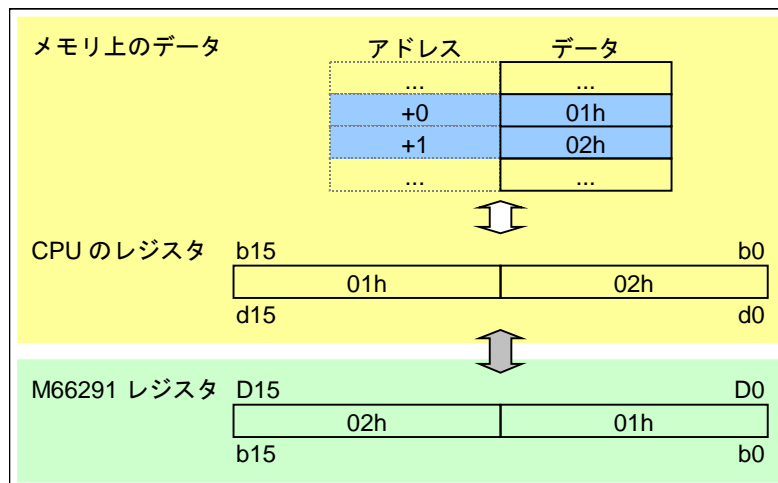


図 7-10 ビットエンディアン:リトル、バイトエンディアン:ビッグのレジスタアクセスイメージ(16bit バス幅)(2)

<FIFO バッファ(FIFO_DATA レジスタ)アクセス>

(1) M66291 の FIFO に対する CPU アクセス

基本的にBSWP(バイトスワップモード;Address 40h, bit 12)=0¹、Octl(レジスタ8bitモード;Address 40h, bit 6)¹=0¹の設定でアクセスします。奇数バイト数書き込み時の最後の1バイトのみ、BSWP='0'、かつOctl='1'を設定し、CPUのD15-D8に有効なデータを、CPUのD7-D0にダミーデータを出力し、16ビット単位で書き込みます。

読み出し: M66291のFIFOバッファに格納された順番で、メモリに格納されます。

書き込み: メモリに格納された順番で、M66291のFIFOバッファに格納されます。

(2) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。但し、BSWP、Octlは、Dn_FIFO選択レジスタ(Address 48h or 50h, bit 12)上のBSWP、Octl²を使用します。

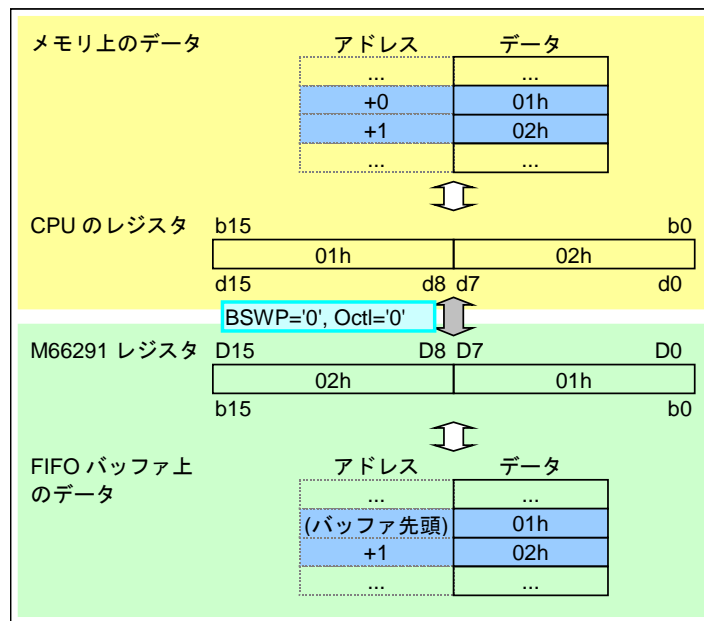


図 7-11 ビットエンディアン:リトル、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(2) (16 ビットアクセス)

¹ Octl(Address 40h, bit 6)と、Epi_Octl(Address (62+(i-1)*2)h, bit 10)(i は CPU_EP(Address 40h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。どちらか一方が'1'に設定されると、当該エンドポイントの FIFO バッファアクセスは 8 ビットモードになります。

² Octl(Address 48h or 50h, bit 6)と、Epi_Octl(Address (62+(i-1)*2)h, bit 10)(i は DMA_EP(Address 48h or 50h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。

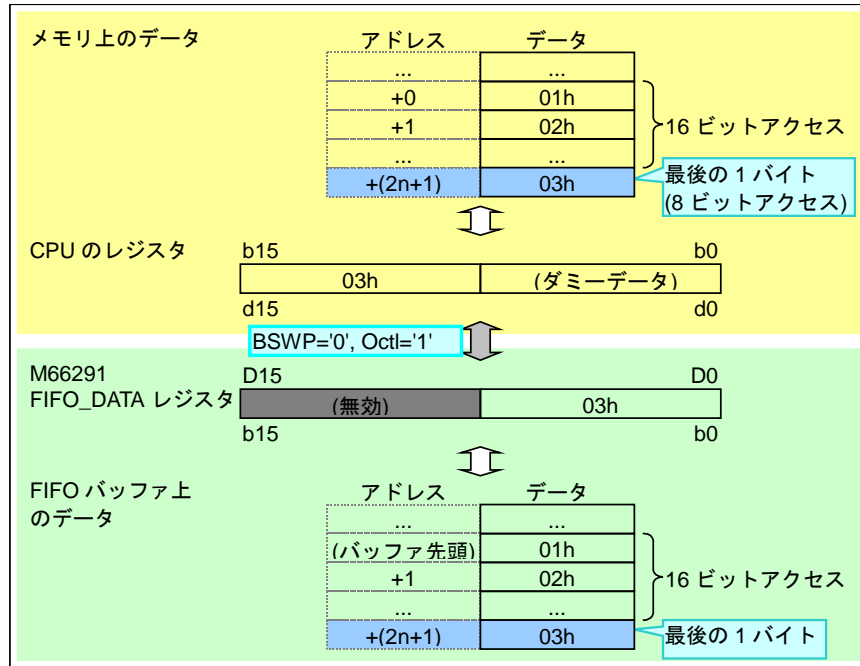


図 7-12 ビットエンディアン:リトル、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(2) (バイト書き込み)

7.1.3. ビットエンディアンがビッグ、バイトエンディアンがビッグの場合(16bit バス幅)

ルネサスの M32R ファミリーなどはこの種類の CPU です。M66291 との接続を考える場合には、基本的には方法(1)(7.1.3.1章)の方法を推奨します。参考資料として方法(2)(7.1.3.2章)を併記します。

7.1.3.1. ビットエンディアンがビッグ、バイトエンディアンがビッグの場合(16bit バス幅) (1)

<接続>

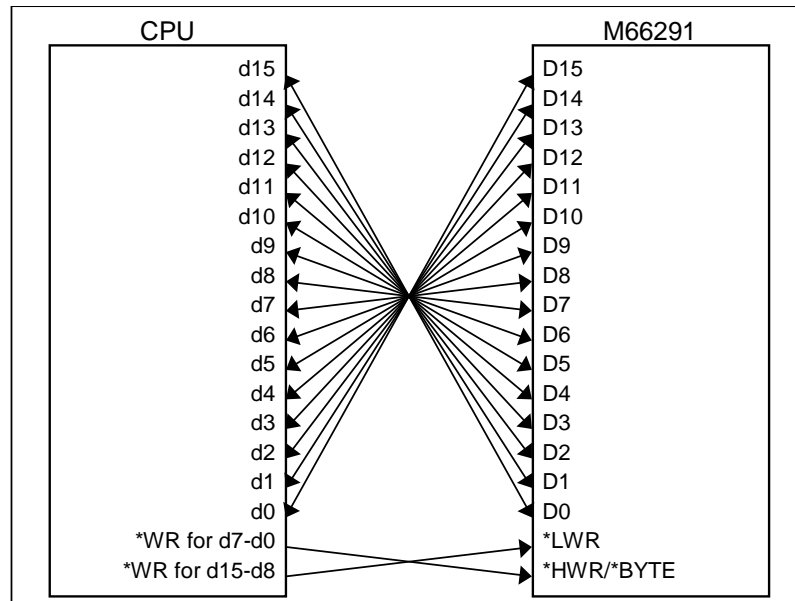


図 7-13 ビットエンディアン:ビッグ、バイトエンディアン:ビッグの接続例(16bit バス幅) (1)

<レジスタアクセス(FIFO_DATA レジスタ以外)>

読み出し:M66291のレジスタのデータがそのまま、CPUのレジスタ或いはメモリに読み出されます。

書き込み:CPUのレジスタ或いはメモリ上のデータがそのまま、M66291のレジスタに書き込まれます。

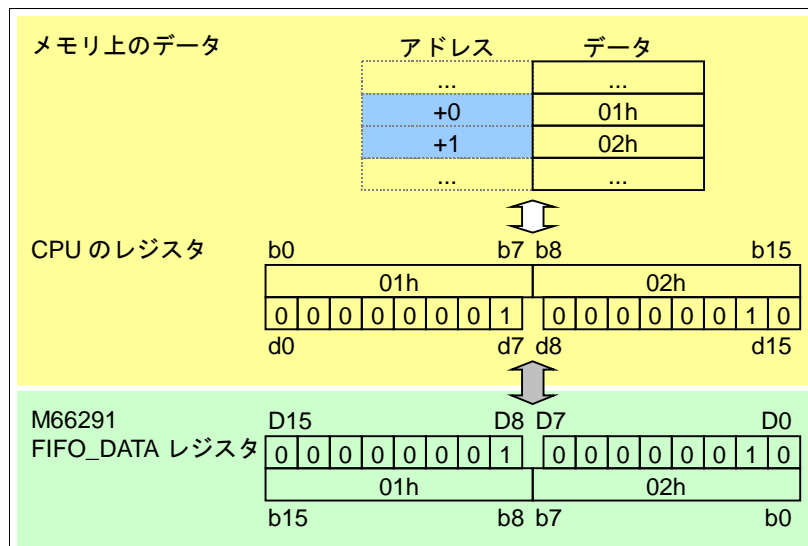


図 7-14 ビットエンディアン:ビッグ、バイトエンディアン:ビッグのレジスタアクセスイメージ(16bit バス幅)(1)

<FIFO バッファ(FIFO_DATA レジスタ)アクセス>

(1) M66291 の FIFO に対する CPU アクセス

基本的にBSWP(バイトスワップモード;Address 40h, bit 12)='1'、Octl(レジスタ8bitモード;Address 40h, bit 6)¹='0'の設定でアクセスします。BSWP='1'の場合、CPU_FIFO-FIFOバッファ間のデータ転送時にM66291内部でバイト単位で上位/下位を入れ替えます。即ち、読み出し時はFIFOバッファの下位バイトがM66291のD15-D8に、上位バイトがM66291のD7-D0に読み出されます。書き込み時は、M66291のD15-D8のデータがFIFOバッファの下位バイトに、M66291のD7-D0のデータが上位バイトに書き込まれます。

奇数バイト数データの最後の1バイトのみ、BSWP='0'、かつOctl='1'を設定し、CPUのD15-D8に有効なデータを、CPUのD7-D0にダミーデータを出力し、16ビット単位で書き込みます。

読み出し: M66291のFIFOバッファに格納された順番で、メモリに格納されます。

書き込み: メモリに格納された順番で、M66291のFIFOバッファに格納されます。

(2) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。但し、BSWP、Octlは、Dn_FIFO選択レジスタ(Address 48h or 50h, bit 12)上のBSWP、Octl²を使用します。

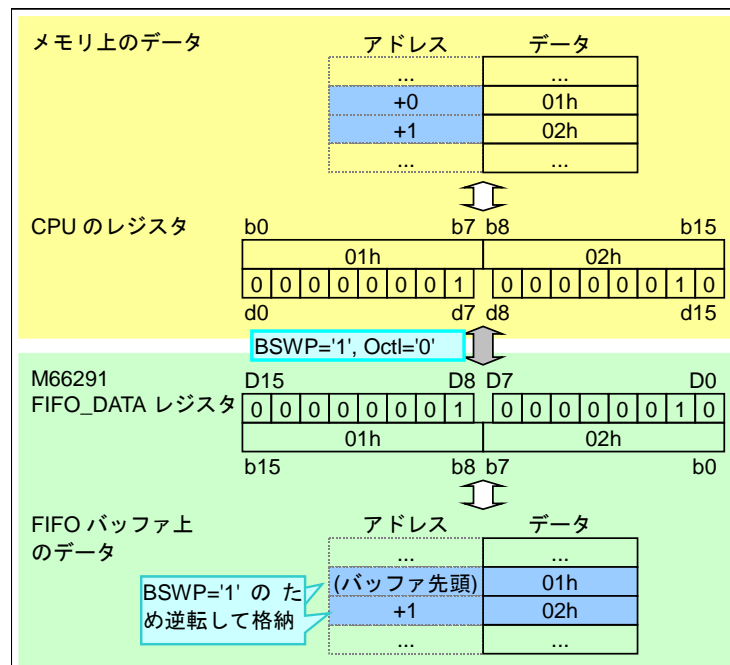


図 7-15 ビットエンディアン:ビッグ、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(1) (16 ビットアクセス)

¹ Octl(Address 40h, bit 6)と、Epi_Octl(Address (62+(i-1)*2)h, bit 10)(i は CPU_EP(Address 40h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。どちらか一方が'1'に設定されると、当該エンドポイントの FIFO バッファアクセスは 8 ビットモードになります。

² Octl(Address 48h or 50h, bit 6)と、Epi_Octl(Address (62+(i-1)*2)h, bit 10)(i は DMA_EP(Address 48h or 50h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。

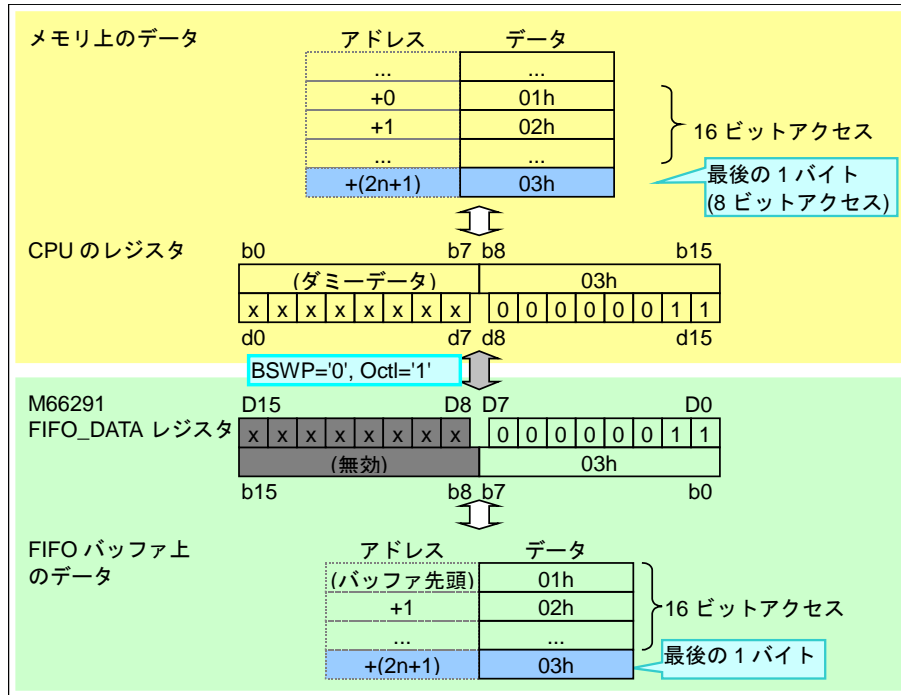


図 7-16 ビットエンディアン:リトル、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(1) (バイト書き込み)

7.1.3.2. ビットエンディアンがビッグ、バイトエンディアンがビッグの場合(16bit バス幅)(2)

<接続>

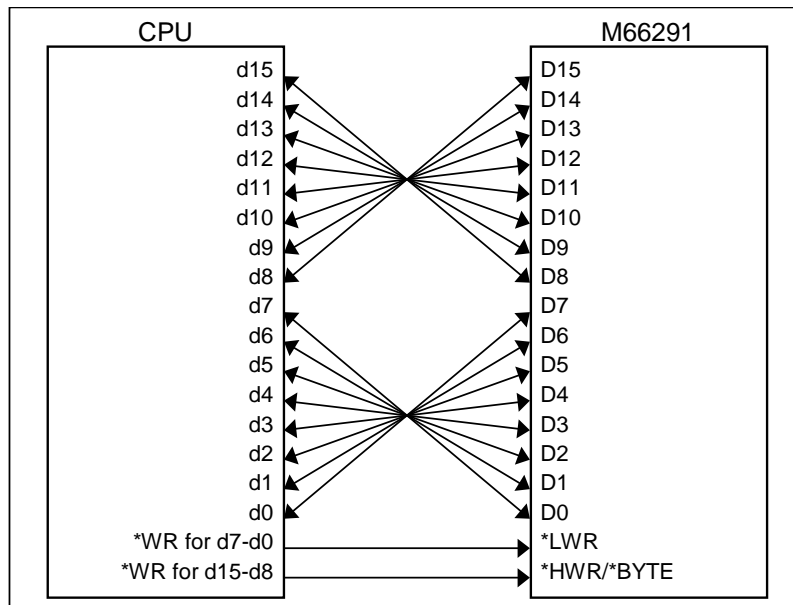


図 7-17 ビットエンディアン:ビッグ、バイトエンディアン:ビッグの接続例(16bit バス幅) (2)

<レジスタアクセス(FIFO_DATA レジスタ以外)>

読み出し: M66291のレジスタのデータがバイト単位で逆転して、CPUのレジスタ或いはメモリに読み出されます。

書き込み: CPUのレジスタ或いはメモリ上のデータがバイト単位で逆転して、M66291のレジスタに書き込まれます。

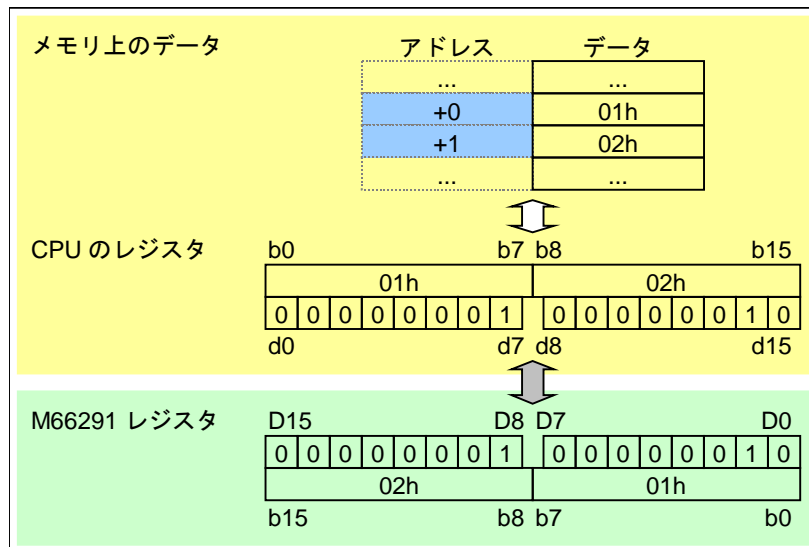


図 7-18 ビットエンディアン:ビッグ、バイトエンディアン:ビッグのレジスタアクセスイメージ(16bit バス幅)(2)

<FIFO バッファ(FIFO_DATA レジスタ)アクセス>

(1) M66291 の FIFO に対する CPU アクセス

基本的にBSWP(バイトスワップモード;Address 40h, bit 12)=**'0'**、Octl(レジスタ8bitモード;Address 40h, bit 6)¹=**'0'**の設定でアクセスします。奇数バイト数書き込み時の最後の1バイトのみ、BSWP=**'0'**、かつOctl=**'1'**を設定し、CPUのd0-d7に有効なデータを、CPUのd8-D15にダミーデータを出力し、16ビット単位で書き込みます。

読み出し: M66291のFIFOバッファに格納された順番で、メモリに格納されます。

書き込み: メモリに格納された順番で、M66291のFIFOバッファに格納されます。

(2) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。但し、BSWP、Octlは、Dn_FIFO選択レジスタ(Address 48h or 50h, bit 12)上のBSWP、Octl²を使用します。

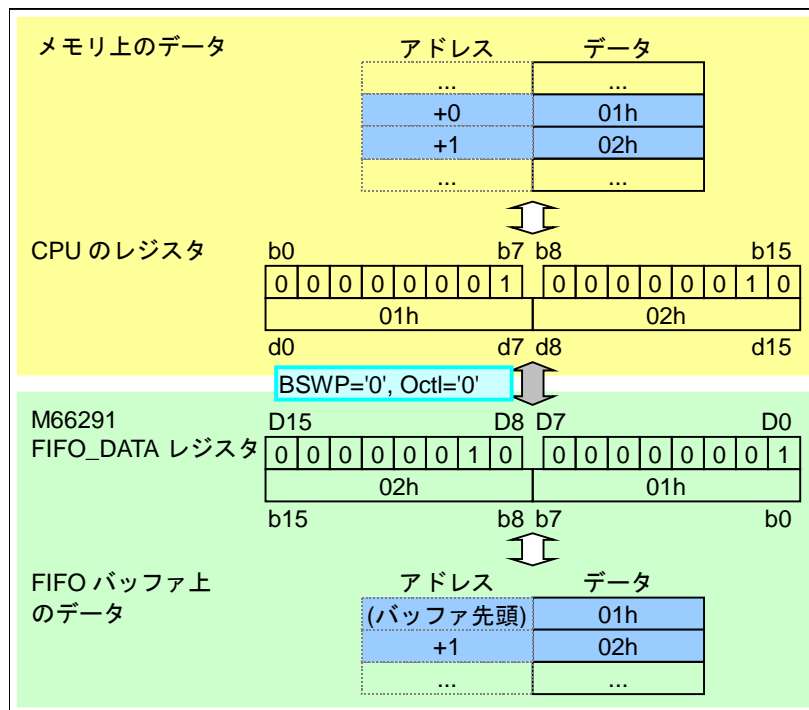


図 7-19 ビットエンディアン:ビッグ、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(2) (16 ビットアクセス)

¹ Octl(Address 40h, bit 6)と、EPI_Octl(Address (62+(i-1)*2)h, bit 10)(i は CPU_EP(Address 40h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。どちらか一方が'1'に設定されると、当該エンドポイントの FIFO バッファアクセスは 8 ビットモードになります。

² Octl(Address 48h or 50h, bit 6)と、EPI_Octl(Address (62+(i-1)*2)h, bit 10)(i は DMA_EP(Address 48h or 50h, bit 3-0)で設定しているエンドポイント)は"or"で反映されます。

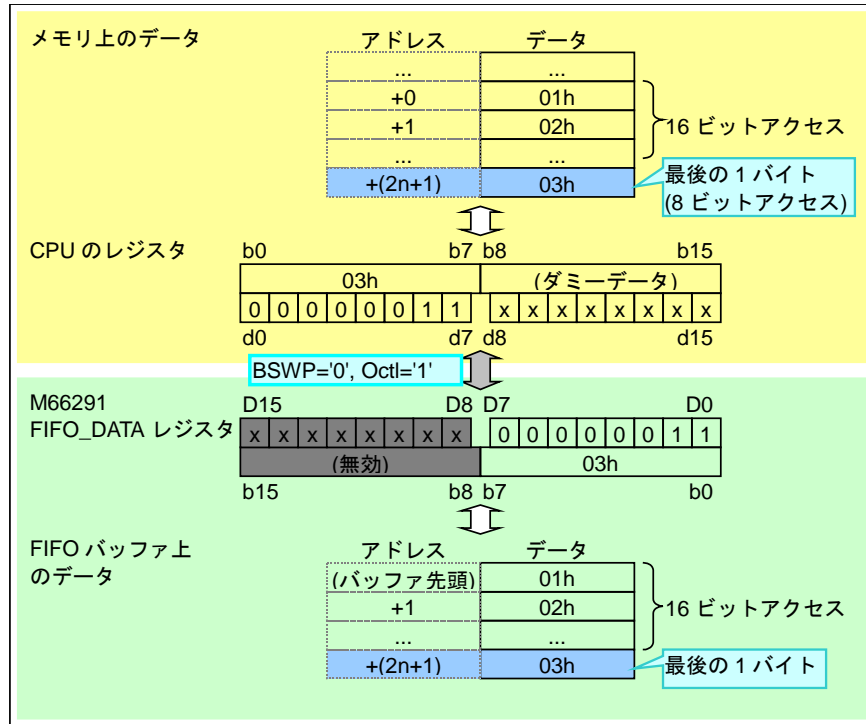


図 7-20 ビットエンディアン:ビッグ、バイトエンディアン:ビッグの FIFO バッファアクセスイメージ(16bit バス幅)(2) (バイト書き込み)

7.2. CPU との接続方法(8bit バス幅)

この章では、8bit バス幅での各 CPU と M66291 との結線方法、及びアクセス方法について説明します。CPU とエンディアンの種類については、表 7-1を参照ください。M66291 へのアクセスは、レジスタへのアクセス、FIFO バッファへのアクセスに大別され、FIFO バッファへのアクセスには、CPU によるアクセスと DMA 転送によるアクセスがあります。

8bit バス幅接続の場合、FIFO バッファへのアクセス方法は接続方法によらず共通となります。FIFO バッファへのアクセスイメージを 7.2.1 外部 8bit バス幅の場合の FIFO バッファアクセスに示します。FIFO バッファへのアクセスは常にバイト単位で行ってください。

16 ビットアクセス可能な CPU と M66291 が 8bit バス幅で接続される場合のレジスタアクセスについては、CPU のエンディアンを考慮する必要があります。レジスタへのアクセスイメージについて7.2.2章-7.2.5章に示します。各種エンディアンの CPU と M66291 の接続性を表 7-2にまとめます。

表 7-3 CPU のエンディアンによる M66291 の接続性(8bit バス幅接続時)

CPU のエンディアン		CPU からの 16bit アクセス可否	接続方法	BSWP 設定値 ²	アクセス ¹	
ビット	バイト				レジスタ	FIFO
リトル	リトル	可の CPU	7.2.2章の方法	'0'	ok	ok
リトル	ビッグ	可の CPU	7.2.3章の方法	'0'	invert	ok
リトル	-	不可の CPU	7.2.4章の方法	'0'	ok	ok
ビッグ	ビッグ	可の CPU	7.2.5章の方法	'0'	invert	ok
-	ビッグ	不可の CPU	検討省略	-	-	-

実際に設計されるときは、お客様の使い方にあわせて結線方法を選択する必要があります。

¹ ok:データのありのまま読み出せる、或いは書き込めることを示します。
invert:バイトが逆転して読み出せる、或いは書き込めることを示します。
² 8bit バス幅接続の場合は、BSWP='0'に固定して使用してください。

7.2.1. 外部 8bit バス幅の場合の FIFO バッファアクセス

(1) M66291 の FIFO に対する CPU アクセス

BSWP(バイトスワップモード;Address 40h, bit 12)=0'の設定でアクセスします。常に8ビットの変数/レジスタを使用し、M66291のFIFOレジスタのb7-b0に対してアクセスしてください(即ち、偶数アドレスのみにアクセスしてください)。

読み出し: M66291のFIFOバッファに格納された順番で、メモリに格納されます。

書き込み: メモリに格納された順番で、M66291のFIFOバッファに格納されます。

(2) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。DMACの転送単位をバイトに指定してください。BSWP(Address 48h or 50h, bit 12) =0'に設定してください。

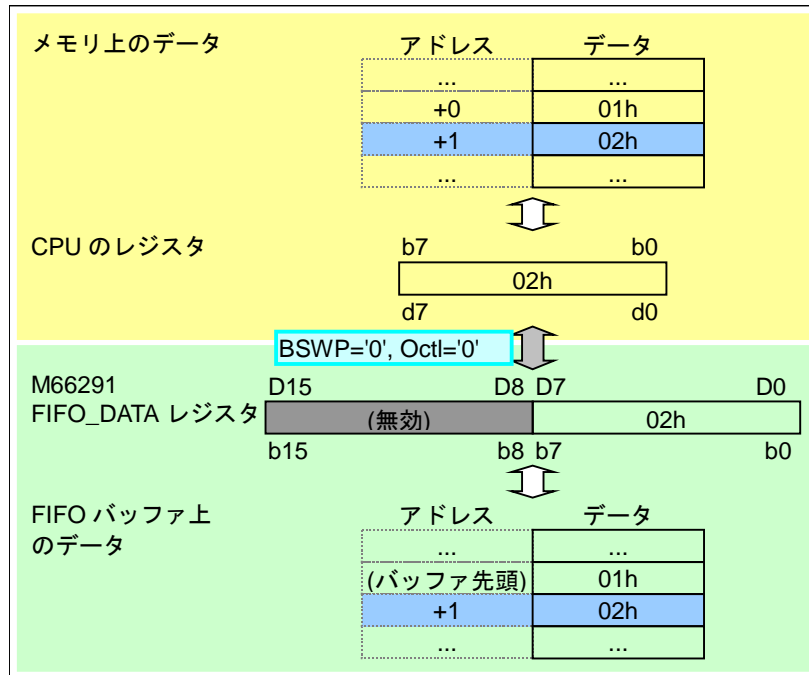


図 7-21 FIFO バッファアクセスイメージ(8bit バス幅)

7.2.2. ビットエンディアンがリトル、バイトエンディアンがリトルの CPU の場合(8bit バス幅)

<接続>

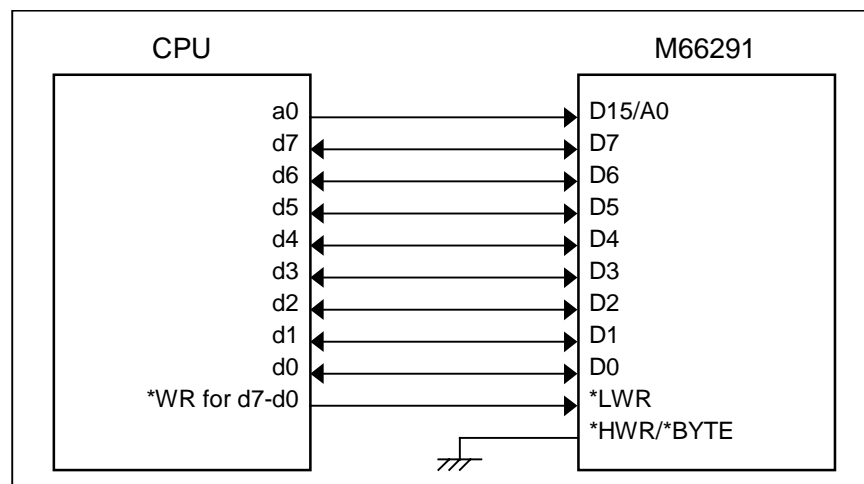


図 7-22 ビットエンディアン:リトルの接続例(8bit バス幅)

<レジスタアクセスイメージ>

FIFO_DATA レジスタ以外のレジスタを CPU 内部で 16 ビットで扱う場合、CPU の下位バイト(D7-D0)が M66291 の偶数アドレスに、CPU の上位バイト(b15-b8)が M66291 の奇数アドレスにアクセスします。

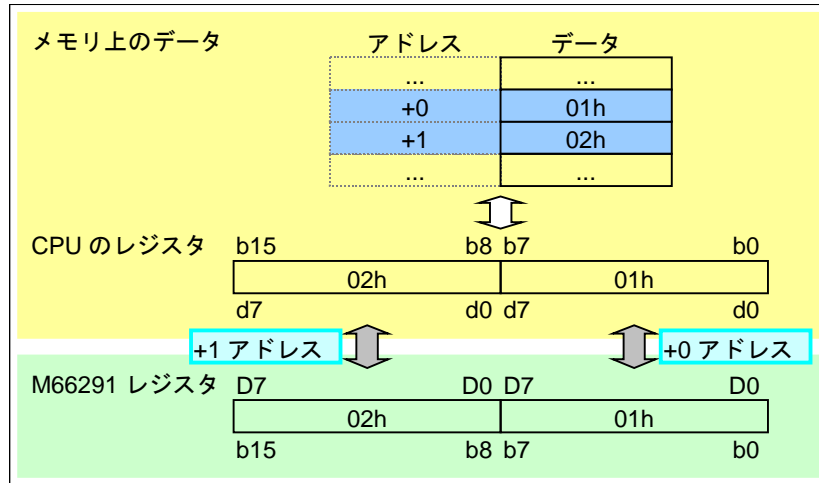


図 7-23 ビットエンディアン:リトル、バイトエンディアン:リトルのレジスタアクセスイメージ(8bit バス幅) (CPU 内部 16 ビット)

7.2.3. ビットエンディアンがリトル、バイトエンディアンがビッグの CPU の場合

<接続>

図 7-22と同様です。

<レジスタアクセスイメージ>

FIFO レジスタ以外のレジスタについて CPU 内部で 16 ビットで扱う場合、CPU の下位バイト(b15-b8)が M66291 の偶数アドレスに、CPU の上位バイト(b7-b0)が M66291 の奇数アドレスに対してアクセスします。

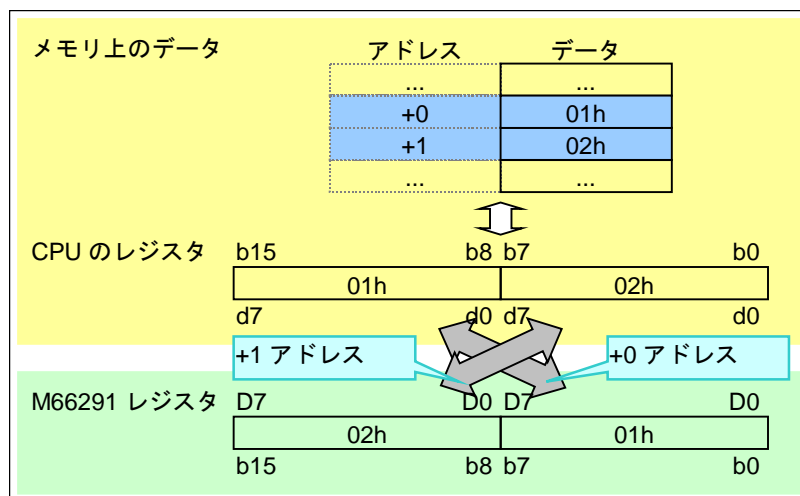


図 7-24 外部 8bit バス幅、ビットエンディアン:リトル、バイトエンディアン:ビッグのレジスタアクセスイメージ(8bit バス幅) (CPU 内部 16 ビット)

7.2.4. 内部 8 ビット、ビットエンディアンがリトルの CPU の場合

<接続>

図 7-22と同様です。

<レジスタアクセスイメージ>

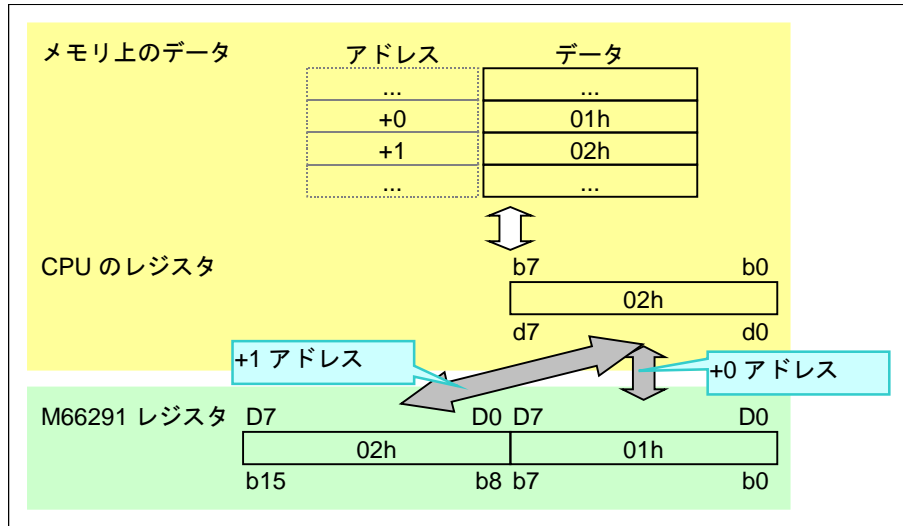


図 7-25 外部 8bit バス幅、ビットエンディアン:リトル、バイトエンディアン:リトルのレジスタアクセスイメージ 1(CPU 内部 16 ビットの場合)

7.2.5. ビットエンディアンがビッグ、バイトエンディアンがリトルの CPU の場合

<接続>

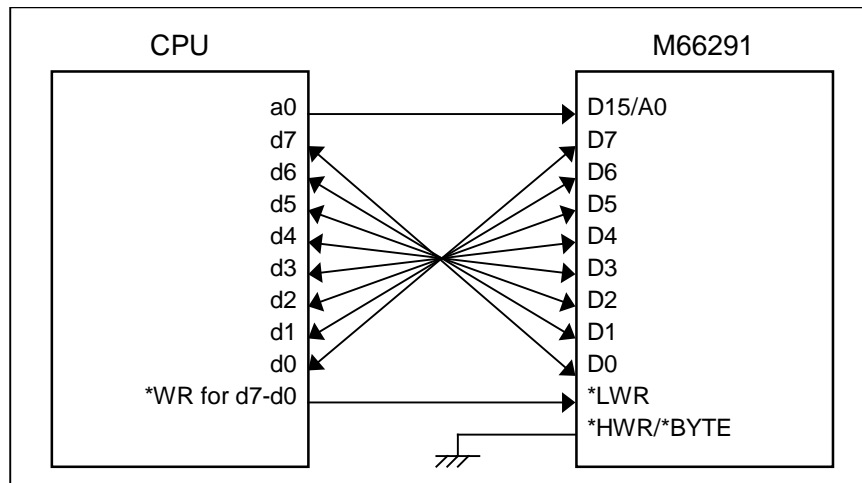


図 7-26 外部 8bit バス幅、ビットエンディアン:ビッグの接続例

<レジスタアクセス>

FIFO レジスタ以外のレジスタについては、CPU の上位バイト(b7-b0)が M66291 の偶数アドレスに、CPU の下位バイト(b15-b8)が M66291 の奇数アドレスにアクセスします。

読み出し: M66291のレジスタのデータがバイト単位で逆転して、CPUのレジスタ或いはメモリに読み出されます。

書き込み: CPUのレジスタ或いはメモリ上のデータがバイト単位で逆転して、M66291のレジスタに書き込まれます。

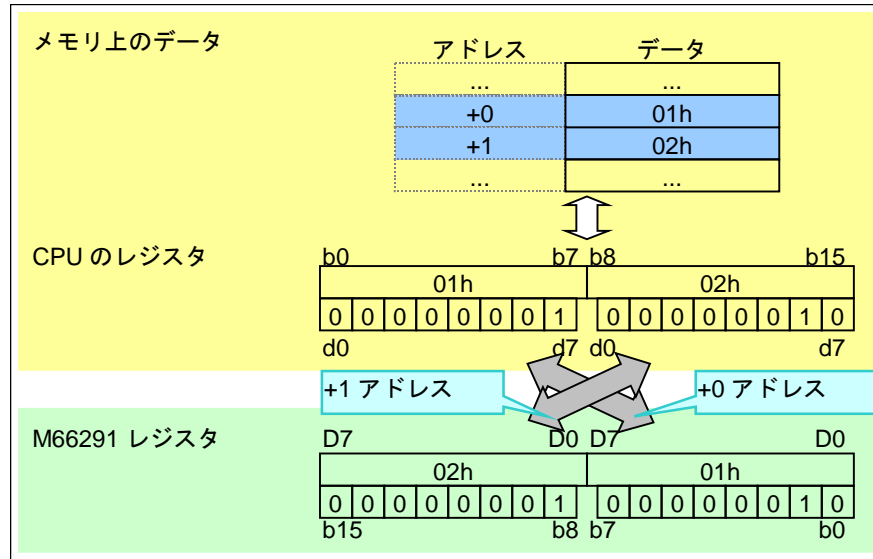


図 7-27 外部 8bit バス幅、ビットエンディアン:ビッグ、バイトエンディアン:ビッグ
のレジスタアクセスイメージ(CPU 内部 16 ビットの場合)

7.2.6. 内部 8 ビット、ビットエンディアンがビッグの CPU の場合

<接続>

図 7-26と同様です。

<データ転送例>

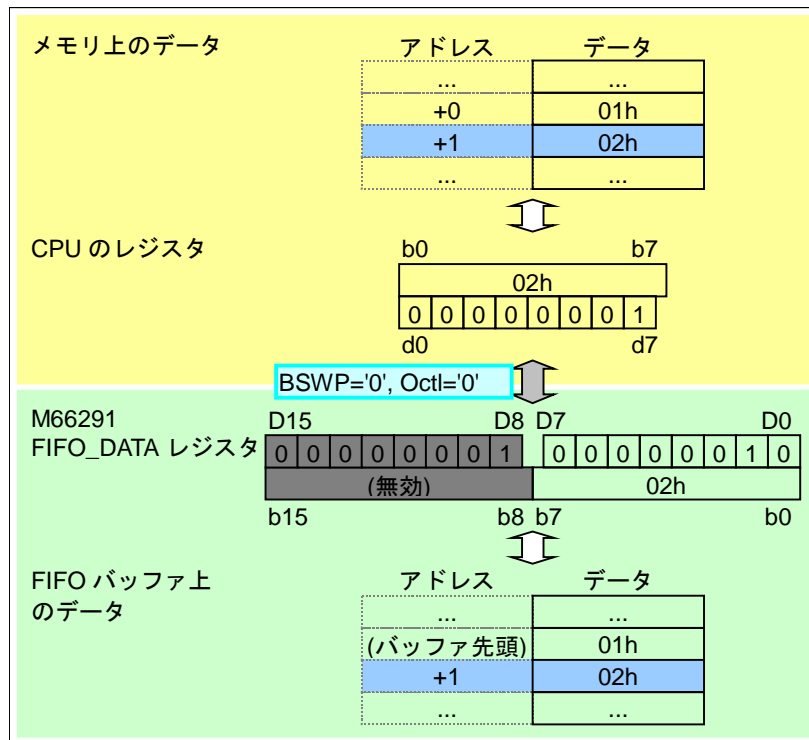


図 7-28 ビットエンディアン:ビッグ、バイトエンディアン:ビッグ
のレジスタアクセスイメージ(CPU 内部 16 ビット)

<アクセス方法>

(1) M66291 のレジスタに対する CPU アクセス

M66291のレジスタのb7-b0に対しては偶数アドレスで、b15-b8に対しては奇数アドレスでアクセスしてください。

(2) M66291 の FIFO に対する CPU アクセス

BSWP(バイトスワップモード;Address 40h, bit 12)=0'の設定でアクセスします。常にM66291のFIFOレジスタのb7-b0に対してアクセスしてください(即ち、偶数アドレスのみにアクセスしてください)。

読み出し: M66291のFIFOバッファに格納された順番で、メモリに格納されます。

書き込み: メモリに格納された順番で、M66291のFIFOバッファに格納されます。

(3) M66291 の FIFO に対する DMA アクセス

CPUアクセスと同様です。但し、BSWP(Address 48h or 50h, bit 12)を設定します。また、DMACの転送単位はバイトに指定してください。

7.3. 発振回路

本回路例は、弊社で作製した基板の発振回路例です。

発振回路定数は、発振子メーカーが、一定条件のもと（指定のない限り、常温で電源電圧は 3.3V）で測定したものであり、実際のシステムにおける高周波、ノイズ、電源特性、配線パターン等の周辺環境については、一切考慮しておりません。したがって、実際のシステムにおいては、値が異なる場合があります。

尚、ルネサス テクノロジ及びルネサス ソリューションズは、この回路定数を保証しておりません。必ず参考値としてご利用下さい。

お客様のシステムにおける最適発振回路定数が必要な場合は、発振子メーカーにご依頼下さい。

また、回路を構成する際に安定化している電源の使用及び、M66291 の電源端子(CoreVcc,IOVcc)と Ground 端子(GND)間にパスコンを入れることを推奨します。

図 7-29 は株式会社大真空製水晶発振子を使用した例です。

図 7-30 は株式会社村田製作所製セラミック発振子を使用した例です。

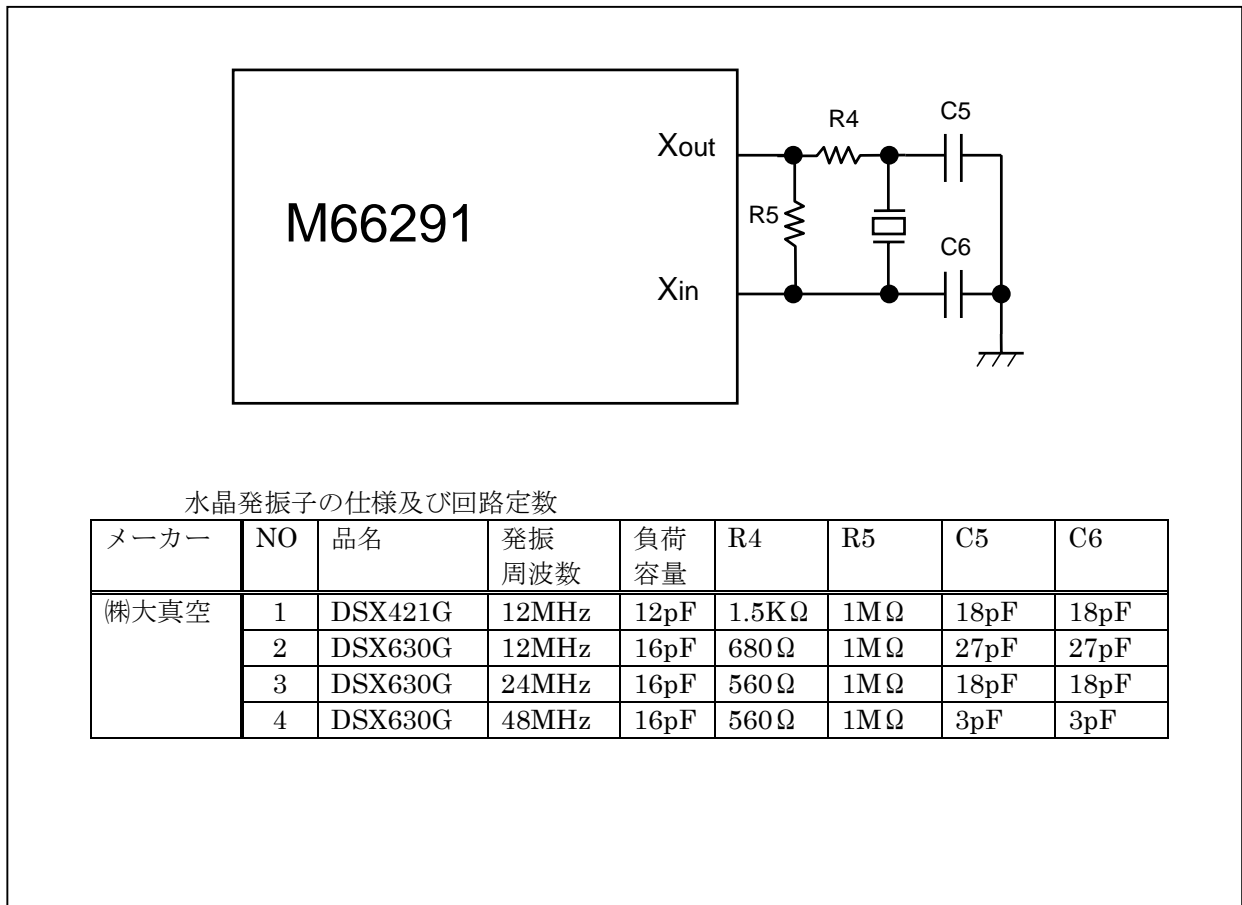


図 7-29 発振回路例 1（株式会社大真空製水晶発振子）

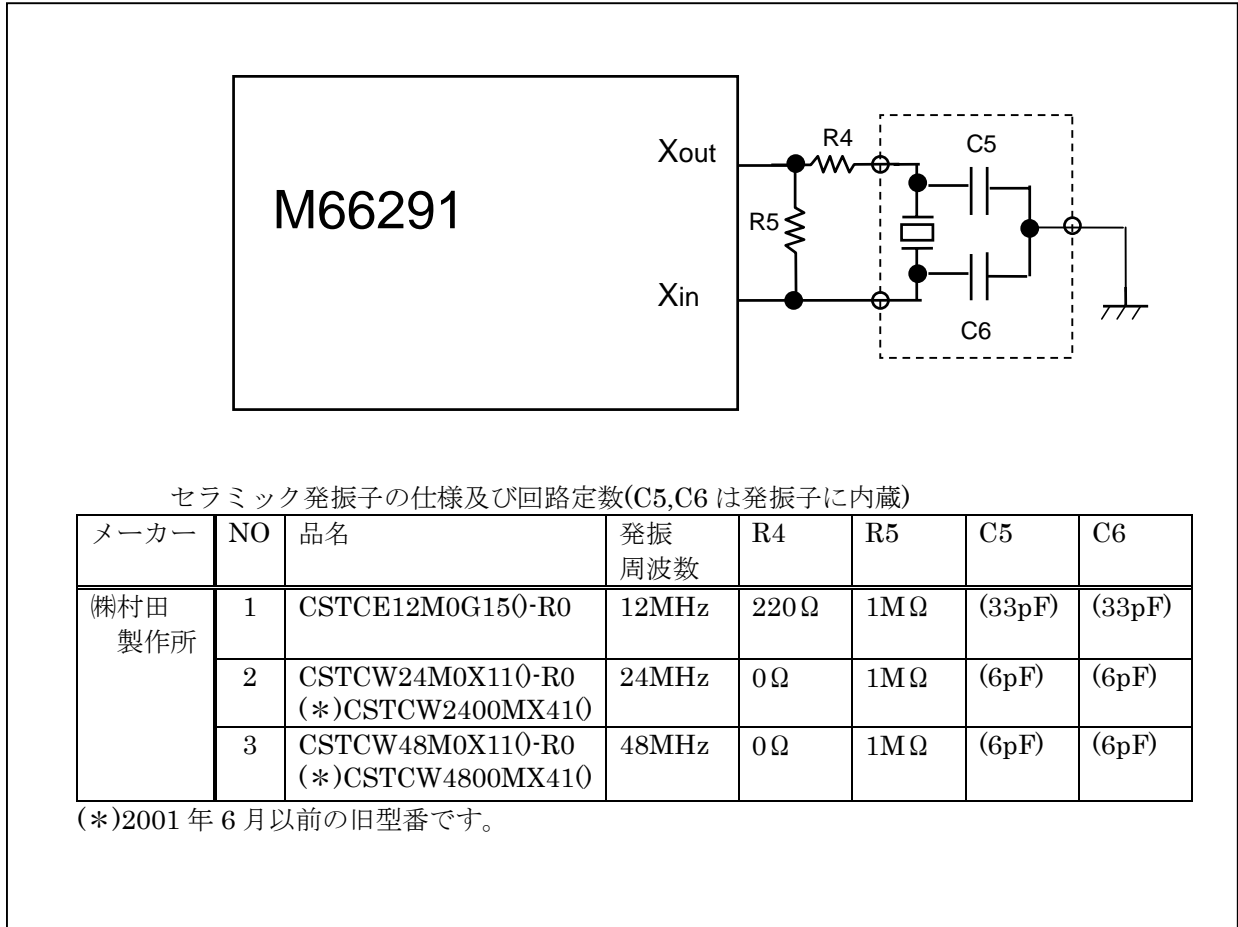


図 7-30 発振回路例 2 (株式会社村田製作所セラミック発振子)

※注意点

上記 NO.2,NO.3 は狭偏差対応タイプです。

USB 用途でご使用される場合、発振周波数精度が厳しいものとなりますので、ご使用される基板（基板が変更する）毎に評価を行う必要がありますのでご注意ください。

発振子の周波数合わせ込み選別により、各基板ごとに品名（ ）内にカスタム品番をメーカーよりつけられることを推奨されております。詳細は発振子メーカーにお問い合わせ下さい。

8. M66291 と M66290A の主な相違点

M66291 と M66290A の主な相違点を以下に述べます。また、本アプリケーションノートの付録として、レジスタの相違点一覧を添付します。

8.1. 追加機能について

8.1.1. USB 転送機能に関する機能追加

8.1.1.1. エンドポイント数増加

エンドポイント 6 が追加され、最大 7 本のエンドポイントが使用可能となりました。

8.1.1.2. コントロール転送の連続転送機能強化

コントロール転送における連続転送使用時の設定可能最大データ数を 255 バイトから 256 バイトに増加しました。

このことにより、255 バイトを超えるデータの送受信時にも、ショートパケットがはさまれませんので、1 回の連続転送設定で転送できるデータ数に実質的に制限がなくなりました。

8.1.1.3. データ転送の連続転送機能強化

OUT 方向の連続転送使用時におけるデータ読み出し機能を追加しました。

受信データ数がマックスパケットサイズの整数倍、かつ、FIFO バッファ容量より小さい場合にも CPU がデータを読み出せるように、SIE_FIFO ステータスレジスタ(Address 46h)、および、トランザクション回数レジスタ(Address 4Eh)を追加しました。

8.1.1.4. isochronous 転送機能強化

isochronous 転送での SOF への同期性をよりとりやすくするため、IDLY[アイソクロナス送信ディレイセットビット](CPU_FIFO 制御レジスタ(Address 42h)、bit 14)を追加しました。また、USB バスの SOF パケット受信時にパルスを出力する機能を追加しました。SOF パルスは、"L"アクティブ/"H"アクティブを選択できます。

8.1.2. バスインターフェース機能に関する機能追加

8.1.2.1. バスインターフェース電源トレラント強化

3.3V 動作、5V 動作の CPU と M66291 を直接接続可能にしました。M66291 へのコア Vcc=3.3V は必要ですが、制御用 CPU、外付け DMA コントローラ等の動作電圧が 5V の場合でも外付けバッファ等は必要ありません。

5V インターフェイスでご使用になる場合は、M66291 の初期化処理で LDRV[駆動電流調整ビット](駆動電流調整レジスタ(Address 5Eh)、bit 0)=1'を設定してください。

8.1.2.2. 8bit バス幅インターフェース追加

8bit バス幅で M66291 を制御することが可能になりました。この場合は、M66291 起動時に *HWR/*BYTE 端子に"L"レベルを入力してください。8bit バス幅で使用される場合、空き端子となる D8-D14 はプログラマブル入出力ポートとして使用できます。

8.1.2.3. バッファレディ割り込みステータスクリア機能追加

M66290A ではバッファレディ割り込みのステータス(割り込みステータスレジスタ 1(Address H'18))クリアはソフトウェアでは行えませんが、M66291 では、RDYM[レディモードビット](極性設定レジスタ(Address 0Eh)、bit 2)=1'に設定すればバッファレディ割り込みのステータスをソフトウェアでクリアすることができます。RDYM=0'設定時(M66290A 互換時)は IN 方向のエンドポイントに関して、送信すべきデータがないタイミングでレディ割り込みが発生したままになるため、回避するために当該エンドポイントのバッファレディ割り込みを禁止する処理が必要ですが、RDYM=1'設定時は他の割り込みと同様にソフトウェアで制御できます。このため、バッファのステータスを示す FIFO ステータスレジスタ(Address 58h)を追加しました。RDYM=0'設定時は、割り込みステータスレジスタ 1 と FIFO ステータスレジスタは同じ内容となります。

RDYM=1'設定時は、OUT 方向に設定したエンドポイントの FIFO バッファの読み出し完了時にバッファクリア(BCLR=1')が必要ですのでご注意ください(RDYM=0'設定は読み出し終了時にハードウェアでクリアされます)。

8.1.2.4. FIFO_DATA レジスタ機能強化

D1_FIFO レジスタ(Address 54h)を追加し、データ転送用として合計 3 つの FIFO_DATA レジスタが使用可能となりました。

また、Dn_FIFO レジスタに対して、Dack ハンドシェイクを使用しないモード、CPU による読み出し/書き込みモード等を追加し、アクセスの利便性を強化しました。

8.1.2.5. 割り込み機能強化

CPU への割り込み出力制御用に**極性設定レジスタ(Address 0Eh)**を追加しました。以下の制御が可能です。

- (1) 割り込み出力端子を最大 2 本使用でき、割り込み要因毎に出力端子を設定できます(SOF パルス出力を使用する場合は*INT0 のみ使用可)。
- (2) 割り込み出力端子の"L"アクティブ/"H"アクティブを選択できます。
- (3) 割り込み出力端子のエッジセンス/レベルセンスを選択できます。

尚、リセット直後は M66291 の割り込み出力端子は"L"アクティブで起動します。"H"アクティブでご使用になる場合、切り替え直後に空割り込み発生の可能性があります。CPU の仕様をご確認の上、処理設計をお願いします。

8.2. M66290A との互換性について

M66290A と M66291 の互換性について以下に説明します。実際の移植時にはお客様にて十分にご検討と評価をお願いいたします。

8.2.1. CPU との接続方法

CPU との接続方法は、[図 2-3](#)を参照ください。

M66290A との互換機能で使用する場合に機能が異なる端子を表 8-1 に示します。

また、FIFO アクセスサイクル高速化に伴い、バスタイミング規定が M66290A と異なっていますので、制御 CPU とのタイミング検証を十分に実施ください。

表 8-1 M66291/M66290A 端子相違点

端子 No.	M66291		M66290A	
	端子名	端子処理	端子名	端子処理
8	*Dack1(入力)	未使用時"H"固定	*TRST(入力)	*RST とショート
9	*Dreq1(出力)	未使用時オープン	TCK(入力)	未使用時"L"固定、または、オープン
10	*TC1(入力)	未使用時"H"固定	TMS(入力)	未使用時"H"固定、または、オープン
11	*INT1/*SOF(出力)	未使用時オープン	TDI(入力)	未使用時"H"固定、または、オープン
12	IOVcc	Vcc とショート	TDO(出力)	未使用時オープン
35	IOVcc	Vcc とショート	Vcc	Vcc
41	*HWR/*BYTE	*LWR とショート	TEST2	"L"固定、または、オープン

8.2.2. ソフトウェア移植時の注意点

M66291 で機能が追加されたレジスタについても、デフォルト値は M66290A と互換動作となるように設計されていますが、例外を以下に示します。

8.2.2.1. バッファリワインド機能

(1) エンドポイント 0 のバッファリワインド機能

M66290 では、ISEL[バッファセレクトビット](EP0_FIFO 選択レジスタ(Address 30h)の bit 0)を変更するとエンドポイント 0 に割り当てた FIFO バッファのポインタを初期化します。

M66291 では、ISEL を変更してもバッファポインタは初期化されません。

(2) エンドポイント 1~5 のバッファリワインド機能

RWND[バッファリワインドビット](CPU_FIFO 選択レジスタ(Address 40h)、Dn_FIFO 選択レジスタ(Address 48h、50h) の bit 12)が追加され、CPU_EP/DMA_EP(同レジスタ b3-b0)の切替時に当該エンド

ポイントにアサインした FIFO バッファのポインタを初期化する/しないをソフトウェアで指定できるようになりました。

M6690A では、CPU_EP/DMA_EP を切り替えると必ずポインタが初期化されます。

M66291 のデフォルト(RWND='0')は、切り替え時のバッファポインタ初期化が発生しません。RWND='1'を設定すると、バッファポインタの初期化が発生します。

(2) DMA インターフェイスの設定

M66290A と M66291 では DMA インターフェイスの設定ビットがビットの位置/機能ともに異なっています。詳細は、データシート及び、[2.7 Dn_FIFO データレジスタへのアクセス\(DMA コントローラインターフェース\)](#)を参照ください。

M66290A の"1 ワード転送モード"が M66291 のサイクルスチールモードに相当します。また、M66290A での"高速転送モード"は M66291 にはありません。

(3) バッファエンプティ割り込みの発生タイミング

ダブルバッファで OUT 転送を行う時のバッファエンプティ割り込みの発生タイミングが異なります。

M66290A では、SIE 側 FIFO バッファのデータ送出完了時に、CPU 側 FIFO バッファの書き込みが完了していなければ、バッファエンプティ割り込みが発生します。

M66291 では、SIE 側 FIFO バッファのデータ送出完了時に、CPU 側 FIFO バッファが空であればバッファエンプティ割り込みが発生します。CPU 側 FIFO バッファに対して書き込み中、または書き込み完了の場合は、バッファエンプティ割り込みが発生しません。

8.2.2.2. リモートウェイクアップ機能について

M66290A はリモートウェイクアップ信号出力後、デバイスステートがアドレスステート(Address state)に遷移します。

M66291 はリモートウェイクアップ信号出力後、サスペンドステートに遷移する直前のステートに遷移します。実質的には、構成ステート(Configured state)に遷移します。

9. Q & A

9.1. 基板設計に関する Q & A

9.1.1.	[質問]	パターン設計上の注意事項はありますか。
	[回答]	(1)D+、D-ラインを、なるべく交差させない様に設計します。 (2)基板材料は、一般的なFR4で特に特別なケアはいりません。 (3)ラインは全体的には0.8mm幅程度でコネクタまでの距離は50mm程度を推奨します。
9.1.2.	[質問]	EMI対策はどのように施すのでしょうか。
	[回答]	信号ライン(D+、D-)にコモンモードチョークコイルをいれることが有効です。ただし、この対策によってD+、D-の信号品質に影響を与えますので、部品選定時には信号品質評価を十分に行ってください。
9.1.3.	[質問]	EMC対策はどのように施すのでしょうか。
	[回答]	VBUSラインとGNDラインにフェラライトビーズをいれ入れることが有効です。GNDライン、VBUSに直列にインダクタを入れる為、直列抵抗が低く、かつインダクタンス特性の急峻な部品の選択が必要です。ただし、これによってD+、D-の信号品質に影響を与えますので、部品選定時には信号品質評価を十分に行ってください。
9.1.4.	[質問]	D+、D-に保護回路を挿入する必要はありますか。
	[回答]	現在の所USB specificationでは保護回路については言及しておりません。しかし、USBは活線挿抜を行う規格ですので、実製品ではESD保護を行っている例が多いようです。具体的には保護ダイオードを挿入するのが一般的ですが、実装によりUSB信号特性に直接影響する可能性があるため、USB用の部品を使用してください。
9.1.5.	[質問]	USBチップへのVBUS供給をUSBケーブルからではなく、基板上で+5.0Vに接続して使用することに問題があるでしょうか。
	[回答]	基本的には問題はありません。ただし、プラグの抜き差しによるVbus割り込み判断は行えません。USB specificationでは、VBUSが供給されている時のみD+をプルアップすると規定されているため、別途プラグの抜き差しを検知しD+プルアップを制御する対策が必要です。
9.1.6.	[質問]	電源電圧2.7V動作は可能でしょうか。
	[回答]	電源電圧(CoreVcc)は3.0V-3.6Vの範囲で使用ください。USB specificationではD+/D-信号の出力レベル(V _{OH})をMin.:2.8Vと規定しています。電源電圧2.7VではこのV _{OH} 規格を満足できません。 電源電圧(CoreVcc)の設定とは関係なく、I/O電源(IOVcc)は2.7V-5.5Vの範囲で動作可能です。
9.1.7.	[質問]	M66291のVbus端子に3.3V入力可能でしょうか。 (USBコネクタVBUS端子から5Vまたは、3.3Vが入力される場合があるため)
	[回答]	USB規格ではデバイス側のVBUSの電圧は5V(4.4~5.25V)と規定されています。M66291はこの規格に準拠しているため、Vbus端子は必ず5V(4.4~5.25V)を入力してください。3.3Vでは正しく動作しません。
9.1.8.	[質問]	Vbus端子には何ボルト以上で何mA以上流し込めばよいでしょうか。
	[回答]	Vbus端子には4.4V以上を入力して下さい。Vbusの消費電流は、TrON出力をD+のプルアップ抵抗に接続している場合に動作時標準2.2mAです。(D+が0Vにドライブされている場合)
9.1.9.	[質問]	回路設計の際、M66291のデータバス、アドレスバス、コントロール信号線をケアする必要がありますか。
	[回答]	M66291のデータバス、アドレスバス、コントロール信号線をプルアップ(或いはプルダウン)することを推奨します。 M66291のデータバスには、M66291の*CSがアサートされていず、かつ、データバスが中間電位の時に貫通電流が流れない対策がされていますが、基本的には、プルアップまたは、プルダウンを推奨します。また、アドレスバス/コントロール信号線は、常に入力状態ですのでプルアップまたは、プルダウンすることを推奨します。

9.1.10.	[質問]	H/Wリセット直後、S/Wリセット直後の端子の状態を教えてください。
	[回答]	<p>以下のとおりです。詳細はデータシートをご参照ください。</p> <p>(1)H/Wリセット直後は以下のとおりです。</p> <p>① 入力;D7~D0、D14/P6~D8/P0、D15/A0、A6~A1、*CS、*LWR、*HWR/*BYTE、*RD、*Dack0、*Dack1、*TC1、Vbus、*RST、Xin、TEST、D+、D- (但し、*CS、*RD、*Dack0、*Dack1、*TC1は非アクティブ)</p> <p>② 出力;*INT0("H")、*INT1/*SOF("H")、*Dreq0("H")、*Dreq1("H")、TrON(Hi-Z)、Xout("H") (端子名直後の()内は、リセット直後の状態を示します)</p> <p>(2)S/Wリセット直後は以下のとおりです。 以下の端子以外は、リセットする前の状態を保持します。</p> <p>① *INT0、*INT1/*SOFは"H"出力に戻ります。 (M66291は、SWリセットにより割り込みステータスをクリアし、割り込み出力をキャンセルします。また、極性設定レジスタもクリアしますので、割り込み出力がデフォルト("L"アクティブ)に戻ります)</p> <p>② XoutはS/Wリセットの影響を受けません。S/Wリセット直前のUSB動作許可レジスタの設定に従います。</p> <p>➤ 入力端子に中間電位を与える、又は、Vbus、TEST、D+、D-以外の入力端子をオープンのままにすると、リーク電流が発生する可能性があります。システム設計の際にはご注意くださいますようお願いいたします。</p>
9.1.11.	[質問]	内部でPull up/down等端子ケアされている端子がありましたら教えてください。
	[回答]	<p>以下のとおりです。詳細はデータシートをご参照ください。</p> <p>(1)Vbus端子;500kΩでPull downされています。</p> <p>(2)TEST端子;50kΩでPull downされています。</p>
9.1.12.	[質問]	電源が入ったままResetを長時間Assertした場合問題はありませんか?
	[回答]	問題ありません。
9.1.13.	[質問]	CPUインタフェース仕様に関して5Vトレラント対応がされていますか?
	[回答]	IOVcc=5Vの場合、CPUインターフェース関連端子は5Vトレラントとなります。

9.2. コントロール転送に関する Q & A

9.2.1.	[質問]	応答PID(EPO_PID)="STALL"、CCPL='1'を設定してステータスステージでSTALL応答した後、EPO_PIDは"STALL"のままですか、それとも自動的に(ハードウェアによって)に"BUF"になりますか。
	[回答]	EPO_PID="STALL"設定しエンドポイント0でSTALL応答した場合は、次のSETUPトークンを受信するまでEPO_PID="STALL"です。新しいSETUPトークンを受信すると、M66291は、EPO_PIDの設定にかかわらずACK応答し、EPO_PID="NAK"設定を行います。
9.2.2.	[質問]	応答PID(EPO_PID)="STALL"を設定した後、ホストがエンドポイント0のClear Feature リクエストを発行すると、M66291はどのように応答しますか。
	[回答]	応答PID(EPO_PID)="STALL"を設定した場合のM66291の動作は9.2.1のとおりです。従って、M66291は、Clear Feature リクエストに対しACK応答した後、自動的にEPO_PID="NAK"となります。
9.2.3.	[質問]	9.2.2の場合、Clear Feature リクエストを受信後、EPO_PID="BUF"、CCPL='1'を設定し、ステータスステージを終了させればよいのでしょうか。
	[回答]	そのとおりです。
9.2.4.	[質問]	ホストPCへのSTALL応答をハードウェアが自動的に行う場合(M66291が応答PID(EPO_PID)="STALL"を設定し、STALL応答する場合)、及び、ソフトウェアで行う場合(デバイス側アプリケーションがEPO_PID="STALL"を書きこみ、STALL応答する場合)のいずれの場合も、EPO_PID="BUF"への復帰は、手動で行うのですか？
	[回答]	いずれの場合も9.2.1の回答内容のとおりとなりますので、必要なタイミングでソフトウェアでEPO_PID="BUF"を設定する必要があります。
9.2.5.	[質問]	コントロールリード転送の場合、コントロールリード連続送信データ長(SDLN)未満のINトークン受信、又は、SDLNを越えるINトークンを受信した場合には、シーケンスエラーが発生しない限り、EPO_PID="BUF"のままと思っていいですか。EPO_PID="NAK"になるのは、次のセットアップステージとと思っていいですか。
	[回答]	そのとおりです。
9.2.6.	[質問]	指定されたエンドポイントが存在しない場合、または、指定されたリクエストに対応していない場合の具体的なSTALL等の処理を教えてください。
	[回答]	データステージ、または、ステータスステージの処理でEPO_PID="SALL"し設定し、STALL応答します。
9.2.7.	[質問]	連続送信モードでコントロールリード連続送信データ長(SDLN)未満のINトークンが発行された場合にどのような応答を行いますか。
	[回答]	コントロールリード転送において、M66291がデータステージからステータスステージに遷移するタイミングは以下の2つのどちらかです。 (1)ショートパケット送信 (2)OUTトークン受信 したがって、SDLN未満のINトークンに続くOUTトークンを受信した場合には、M66291はステータスステージに遷移します。
9.2.8.	[質問]	連続送信モード時に、EPO_PID="BUF"設定は1回のみですかパケット単位ですか。
	[回答]	1回のみです。1度 EPO_PID="BUF"を設定したら、次のSETUPトークンを受信するまで再設定の必要はありません。
9.2.9.	[質問]	連続送信モード時のEPO_PIDはいつ"NAK"にリセットされるのですか。
	[回答]	コントロール転送で自動的にEPO_PID="NAK"に設定されるのはSETUPトークンを受信した場合だけです。
9.2.10.	[質問]	連続受信モードのコントロールライト連続受信データ長レジスタがありませんが、256バイトの連続受信またはショートパケット受信でしかバッファレディ割り込みは発生しないのですか。
	[回答]	連続受信モードに設定した場合、M66291は、256バイトのデータを受信するかショートパケットを受信するか、INトークンを受信(ホストがステータスステージへ遷移したことを示す)した場合にバッファレディ割り込みを発生させます。

9.2.11.	[質問]	連続受信モードに設定し、FIFOバッファサイズを256に設定し、最大パケットサイズの packets を受信し、かつ、総受信データ数が256バイト未満の場合、割り込みは発生しますか。
	[回答]	連続受信モード設定時のコントロールライト転送におけるバッファレディ割り込み発生については、9.2.10のとおりです。 また、M66291がデータステージからステータスステージに遷移するタイミング(INトークンの受信)で、M66291はコントロール転送ステージ遷移割り込み、及び、バッファレディ割り込みを発生させます。なお、データステージで受信したデータ数と、SETUPステージでホストから指示された受信予定バイト数が一致するかどうかの判定はM66291のハードウェアでは行いません。
9.2.12.	[質問]	連続受信モード設定時は、FIFOバッファサイズを越えて受信されるのですか。
	[回答]	連続受信モード設定時は、エンドポイント0は最大256バイトのバッファサイズを使用します。このため、連続受信モードを設定する場合には、256バイトのFIFOバッファをコントロールライト転送用に割り当てる必要があります。コントロールライト転送のデータステージにおいて、256バイトを超えるデータを受信する場合は、FIFOバッファに蓄積された256バイトのデータをデバイス側アプリケーションが読み出し終わるまで、M66291はホストに対して自動的にNAK応答を行います。
9.2.13.	[質問]	コントロールライト転送で、エンドポイント0のバッファレディ割り込みは、EP0_FIFO選択レジスタをコントロールライトに設定(ISEL='0')したときに発生するのですか。
	[回答]	コントロールライト転送でバッファレディ割り込みが発生するのは、ISEL='0'に設定し、かつ、エンドポイント0のバッファレディ割り込みを許可し、以下(1)、(2)いずれかの状態になった場合です。 (1) データステージでショートパケットを受信した (2) データステージでFIFOバッファがフルになった (3) FIFOバッファに受信データが存在する状態で、M66291がデータステージからステータスステージに遷移した ISEL='1'のを設定した後に、上記(1)(3)いずれかの状態になり、その後ISEL='0'に変更した場合、ISELの値を変更した時点で割り込みが発生します。
9.2.14.	[質問]	エンドポイント0のバッファエンプティ割り込みは、EP0_FIFO選択レジスタをコントロールリード(ISEL='1')に設定したときに発生するのですか。
	[回答]	転送方向を設定した時点では発生しません。バッファエンプティ割り込みの発生条件はFIFOバッファ上のデータ送信が完了したときです。ただし、コントロール転送の場合データステージのみが対象になります。コントロールライト(ISEL='0')転送時のステータスステージにおけるzero-lengthパケット送信ではバッファエンプティ割り込みは発生しません。

9.3. データ転送に関する Q & A

9.3.1.	[質問]	OUT方向の転送で、最大パケットサイズを超えた数のデータ受信した場合に、バッファエンプティ/サイズオーバーエラー割り込みが発生しますが、この場合、受信したデータはFIFOバッファに入っていますか。ソフトウェアでエラーデータを破棄する処理は必要ですか。
	[回答]	ハードウェアで自動的にクリアしますので、ソフトウェアでクリアする必要はありません。
9.3.2.	[質問]	OUT方向の転送で、最大パケットサイズを超えた数のデータ受信した場合にバッファレディ割り込みは発生しますか。
	[回答]	データなしのため、バッファレディにはなりません。
9.3.3.	[質問]	OUT方向の転送で、バッファレディ割り込み発生時に、割り込みステータスをクリアしてもバッファを読み出さないと再度バッファレディ割り込みが発生しますか。
	[回答]	RDYM[バッファレディモードビット]='0'(M66290A互換設定)を設定している場合、バッファレディ割り込みステータスは、ソフトウェアからはクリアできません。OUT方向の場合、バッファフルまたはショートパケットの受信でバッファレディ割り込みが発生し、FIFOバッファからデータをすべて読み出さない限りバッファレディ割り込みが発生したままになります。IN方向の場合、FIFOバッファにデータの書きこみが可能な状態(空の状態)でバッファレディ割り込みが発生します。FIFOバッファに送信データを書き込まない限り、バッファレディ割り込みが発生したままになります。このような場合は、当該エンドポイントのバッファレディ割り込みを禁止して使用する必要があります。 RDYM='1'を設定している場合は、バッファレディ割り込みステータスをソフトウェアでクリアすることができます。(8.1.2.3参照)
9.3.4.	[質問]	応答PID(EPI_PID)="NAK"に設定すると、バッファノットレディ割り込みが発生しないのでしょうか。
	[回答]	応答PID(EPI_PID)="NAK"に設定した場合は、IN/OUTトークンに対してNAK応答した場合でもバッファノットレディ割り込みは発生しません。ただし、コントロール転送におけるセットアップトークンに関しては、EPO_PIDの設定にかかわらずM66291はリクエストを自動的に受信してACK応答します。この場合にもバッファノットレディ割り込みは発生しません。
9.3.5.	[質問]	データ送信(IN方向転送)の場合、送信が終了しホストからACKを受信した時点でM66291はバッファレディ割り込みを発生させる思うのですが、その後バッファノットレディ割り込みを発生させるタイミングはどこでしょうか。
	[回答]	バッファノットレディ割り込みは、当該エンドポイントの応答PID(EPI_PID)="BUF"を設定しているときに、M66291がホストから当該エンドポイントに対するINトークンを受信しNAK応答したときに発生します。
9.3.6.	[質問]	エンドポイントに対するSTALL解除は応答PID(EPI_PID)="BUF"を設定するだけで良いのでしょうか。
	[回答]	そのとおりです。該当PID="BUF"とすることにより、IN/OUTトークン発生時にM66291~ホスト間の通信が可能となります。

9.4. DMA 転送に関する Q & A

9.4.1.	[質問]	OUT方向のBULK転送のDMA使用時にバッファレディ割り込みが発生した場合、ショートパケットを受信したと判断して良いのでしょうか。
	[回答]	INTM='0'に設定している場合、M66291はショートパケットを受信したことでバッファレディ割り込み発生させます。最大パケット受信やFIFOバッファフル毎でのバッファレディ割り込みは発生させず、最終パケットがショートパケットの場合のみ、最終パケットデータのDMA転送完了後にバッファレディ割り込みを発生させます。デバイス側アプリケーションでは、バッファレディ割り込み処理の中で受信データ長を読み出し、DMA転送容量を再設定してください。 INTM='1'に設定している場合(CPUアクセスで転送を行なう場合)は、ショートパケット受信もしくはFIFOバッファフルでバッファレディ割り込みが発生します。
9.4.2.	[質問]	9.4.1 の時DMA_DTLNで受信データ長を判断すれば良いのでしょうか。
	[回答]	そのとおりです。DMA_DTLNをご使用ください。

9.5. FIFO 制御に関する Q & A

9.5.1.	[質問]	FIFOバッファはどのようにクリアするのですか。
	[回答]	FIFOバッファはCPU_FIFO制御レジスタあるいはDn_FIFO制御レジスタの[バッファクリアビット(BCLR)='1'書き込みでクリアします。BCLR='1'でクリアできるバッファは、CPU側バッファだけで、クリアした後にバッファのトグル動作は行いません。 ダブルバッファを設定している場合の、SIE側バッファのクリアはCPU_EPで指定しているIN方向のエンドポイントに対してのみ行うことができます。SCLR[バッファクリアビット]='1'を書き込むことで、CPU_EPに指定しているエンドポイントのSIE側バッファをクリアします。OUT方向のエンドポイントに対してはSCLR='1'を行わないでください。 また、SetConfigurationリクエスト、SetInterfaceリクエスト受信時にはFWでエンドポイントの設定を行いますが、このとき該当EPIコンフィグレーションレジスタ1のACLR='1'、ACLR='0'を連続書き込みすることによりFIFOバッファの初期化を行ってください。
9.5.2.	[質問]	FIFOバッファが完全に空の状態でないときホストからのデータパケット受信はできないのですか。
	[回答]	連続送受信モードに設定した場合は、割り付けたFIFOバッファがフルになるまで、あるいはショートパケットを受信するまで連続してデータパケットを受信できます。非連続送受信モード設定/連続送受信モード設定にかかわらず、ダブルバッファモードに設定すれば、1面のバッファが完全に空きではなくても、もう一方の面が空であれば連続してデータパケットを受信できます。
9.5.3.	[質問]	ショートパケットを受信した場合もFIFOを1面分使用するのですか。
	[回答]	ショートパケットの場合もFIFOバッファは1面分使用します。仮に連続送受信モードに設定し、FIFOバッファを1Kバイトダブルバッファに指定した場合は、ショートパケットを受信することで片面(1Kバイト分)の領域を使用します。
9.5.4.	[質問]	IN方向転送時、M66291は、送信済みのFIFOバッファ上のデータをホストからのACKを受信するまで保持しますか。
	[回答]	保持します。
9.5.5.	[質問]	M66291のハードウェアリセット、又は、内部クロック停止時のレジューム検出から、M66291のレジスタ及びFIFOバッファへアクセス可能になるまでに必要な時間はどの程度ですか。発振安定時間及びPLL-Lock時間等を考慮したWarm up時間を教えてください。
	[回答]	(発振子の発振安定時間)+(M66291内部PLL発振安定時間)を待って、アプリケーションからのアクセスを開始してください。 (1)発振子の発振安定時間はご使用になる発振子に必要な安定時間をとってください (2)M66291内部PLL発振安定時間は約1msです。 M66291内部クロック設定の具体的な手順については、図 3-1、図 3-2図 6-3を参照ください。
9.5.6.	[質問]	ダブルバッファ連続送受信モードに設定してOUT方向転送(受信)するとき、受信途中のFIFOバッファにたまったデータを読み出すことは可能ですか。
	[回答]	TGLビットを使用して、ダブルバッファ連続送受信モード設定時に、受信途中のFIFOバッファ(バッファフルになっていない状態)を読み出すことができます。具体的な手順については、2.7.4 連続転送モード時に、FIFOバッファ容量未滿かつ最大パケットサイズの整数倍のデータを読み出す方法を参照ください。
9.5.7.	[質問]	CPU_FIFO選択レジスタのCPU_EPを切り替える場合の条件を教えてください。
	[回答]	OUT方向のエンドポイントからOUT方向のエンドポイントに変更する場合は、一旦エンドポイント指定をクリア(CPU_EP="0000"と設定)してから、変更先のエンドポイントに変更して下さい。

9.5.8.	[質問]	連続送受信モードで、IN方向転送を行う場合に、プログラムでIVAL='1'を書き込むタイミングを教えてください。
	[回答]	連続送受信モードの場合、書き込みデータ数がFIFOバッファサイズと一致した場合に自動的にIVAL='1'になります。最大パケットサイズではありませんので、ご注意ください。従って、送信データ総数がFIFOバッファサイズの整数倍ではない場合、最終データを書き込みした後、SWでIVAL='1'を書き込んでください。IVAL='1'を書き込むことにより、FIFOバッファは送信待ち状態になります。
9.5.9.	[質問]	コントロール転送では連続送信データ長レジスタがありますが、エンドポイント1~エンドポイント6にはありません。どのようにして送信バイト数を認識するのですか。
	[回答]	エンドポイント1~エンドポイント6では、FIFOバッファに書き込んだバイト数が送信バイト数になります。
9.5.10.	[質問]	連続送受信モードのOUT方向転送において、最大パケットサイズを受信してもFIFOバッファサイズに満たない場合、バッファレディ割り込みは発生しないのですか。もしそうなら、CPUは連続転送の終了をどのようにして認識しますか。
	[回答]	受信したパケットのサイズが最大パケットサイズの場合で、設定したFIFOバッファサイズに満たない場合はバッファレディ割り込みが発生しませんので、以下の2種類のケース以外では、付加的な読み出し処理が必要です。処理方法については、2.6.1、2.7.4を参照ください。 (1) ホストからの転送データが必ずショートパケット(zero-lengthパケットを含む)で終了する。 (2) ホストからの転送データがバッファサイズの整数倍である
9.5.11.	[質問]	EP0_FIFOの読み出し、書き込みを行うときは、バッファセレクト(ISEL)を設定する必要がありますか。
	[回答]	設定する必要があります。
9.5.12.	[質問]	FIFOアクセスサイクル時間tw(CYCLE)の定義は?
	[回答]	CPUアクセスの場合は*WR、*RDのエッジからエッジまでの意味で使用しています。

9.6. zero-length パケット(Null パケット)送信自動付加機能に関する Q & A

9.6.1.	[質問]	<p>FIFOバッファフルまで書き込むと自動的にzero-lengthパケット(Nullパケット)が付加されますか？</p> <p>[例]FIFOバッファ容量128byte設定時に192byte転送すると、以下のシーケンスとなりますか？</p> <p>[2データパケット(1~128バイト目)]送信 [zero-lengthパケット]送信 [1データパケット(129~192バイト目)]送信 [zero-lengthパケット]送信</p>
	[回答]	<p>回答の前提として、連続送受信モード(EPI_RWMD='1')、zero-lengthパケット自動送信モード(EPI_NULMD='1')に設定している場合にzero-lengthパケット自動送信が有効です。</p> <p>シングルバッファ設定時では、ご質問内容どおりのシーケンスになります。</p> <p>ダブルバッファ設定時では、2面目のデータを書き込み終わるタイミングに依存して、zero-lengthパケットが自動送信されるタイミングが異なります。</p> <p>(1) [2データパケット(1~128バイト目)]の送信終了が、残り64バイト(129~192バイト目)の書き込み完了よりも早い場合、ご質問内容どおりのシーケンスになります。</p> <p>(2) [2データパケット(1~128バイト目)]の送信終了が、残り64バイト(129~192バイト目)の書き込み完了よりも遅い場合、以下のシーケンスとなります。</p> <p>[2データパケット(1~128バイト目)]送信 [1データパケット(129~192バイト目)]送信 [zero-lengthパケット]送信</p>
9.6.2.	[質問]	<p>zero-lengthパケット(Nullパケット)自動送信モードはどのような時に使うのでしょうか。</p>
	[回答]	<p>バルク転送でショートパケットの送信によってデータ送信終了をホストに通知するアプリケーションの場合に、連続送受信モードとzero-lengthパケット自動送信モードを使用して、zero-lengthパケットを最後に送信することが出来ます。</p>

9.7. 通信 Error に関する Q & A

9.7.1.	[質問]	受信データ異常の場合(PID異常、CRCエラー、EOP検出エラー等)のデータフィールドの廃棄処理はUSBチップ側で自動的に行いますか?(接続マイコン側は介在しなくて良いですか?)
	[回答]	そのとおりです。ホストからのデータパケットが異常の場合、M66291は異常を検出した時点で受信データを破棄しますのでFW処理は不要です。
9.7.2.	[質問]	CRCの生成/受信CRCのチェックはUSBチップが自動的に生成するのですか?
	[回答]	そのとおりです。

9.8. その他の機能に関する Q & A

9.8.1.	[質問]	OUT方向の転送(M66291が受信)の際、FIFOバッファフルの状態以外でホストに対してNAKを返すことが出来る設定(動作モード)がありますか?
	[回答]	各エンドポイントの応答PID="NAK"、又は、"STALL"に設定することによって、FIFOバッファフル状態以外でホストに対してNAKあるいはSTALLを返すことができます。
9.8.2.	[質問]	消費電流を下げる為、USBを利用していない場合はクロックを停止する予定ですが、その場合、外部からのUSB接続やコマンドの受信により、クロック発振を開始して、USB通信をできるモードに自動で遷移してくれるのでしょうか?
	[回答]	「USB動作許可レジスタ」の操作によってクロックの発振停止やUSBトランシーバ動作禁止ができます(消費電流を下げるができます)。この場合は、USB接続の発生を検出時にM66291がUSB通信を行える状態に戻すためのFWの処理が必要です。具体的な手順は、低消費電力モード移行手順に関しては6.2 サスペンド信号処理を、復帰手順に関しては6.3 レジューム (バス アクティビティ検出)処理を参照ください。
9.8.3.	[質問]	応答PID(EPI_PID)の設定をM66291が変更するケースを教えてください。
	[回答]	エンドポイント1~6の応答PID(EPI_PD)の切り替えに関しては、OUT方向設定で最大パケットサイズ以上のデータパケットを受信した場合のみ、M66291が自動的に応答PIDをSTALLに変更します。このケース以外は、すべてプログラムで切り替えを行う必要があります。
9.8.4.	[質問]	リセットから立ちあがってきたときに、すでにVbus端子が”H”だった場合に、Vbus変化検出割り込みは発生しますか。
	[回答]	Vbus変化検出割り込みは、Vbus端子入力の立ち上がり/立ち下がりを検出します。このため、USB許可(USBE='1')を書き込む前の変化をM66291が検出することはできません。Vbus端子="H"の状態、電源投入、又は、HWリセットを行い、USB不許可状態(USBE='0')から許可状態(USBE='1')にした場合、M66291はVBUS割り込みを発生させませんので、Vbus端子のステータス(Vbusビットの値)で状態を判定してください。
9.8.5.	[質問]	D+プルアップ抵抗制御機能は、ケーブルの活栓挿抜をレジスタでコントロールできる機能ですか。(USBケーブルが接続された状態で、ホスト/Hubに「USBケーブルが抜かれた」と思わせることができますか。)
	[回答]	そのとおりです。USBケーブルが接続された状態でTr_on='01'を書き込むと、M66291はD+信号のプルアップを止めますので、アップストリーム側(Hub)から見るとケーブルが抜かれた状態に見えます。
9.8.6.	[質問]	Vbus割り込みを禁止(VBSE='0')した状態でUSBケーブルを挿抜し、後でVbus割り込みを許可(VBSE='1')すると、Vbus割り込みは発生しますか。
	[回答]	どちらの場合も、Vbus割り込みステータス(VBUS)が1になります。VBUS='0'を書き込まない限り、Vbus割り込みを許可すると割り込み信号が発生します。
9.8.7.	[質問]	外部クロック入力を使用する場合でもUSB動作許可レジスタの発振バッファイネーブルビットを発振許可(XCKE='1')にする必要がありますか。
	[回答]	外部クロック入力を使用する場合でも、XCKE='1'にする必要があります。
9.8.8.	[質問]	低消費電力モードを実現するために、USB通信が行われていない時に本体のCPUはスリープモードになります。これに対応するためにUSBからのパケット受信による割り込み発生ができるようになっていますが、その仕様を教えてください。
	[回答]	USBバスがサスペンド中に、M66291の消費電力を小さくするためにM66291の内部クロック供給を停止していても、USBバスにバス アクティビティが発生するとM66291はレジューム割り込み(RESM割り込み)を発生させます。詳細については、6.2 サスペンド信号処理、6.3 レジューム (バス アクティビティ検出)処理、及び、USB SpecificationのChapter7、Chapter9を参照ください

9.8.9.	[質問]	USBチップへのスリープ解除後の応答に時間制限があるのでしょうか。
	[回答]	(1) サスペンド状態からレジュームされる場合ホスト/Hubは少なくとも20msのレジューム信号を送出しますので、20ms以内に応答の準備を整える必要があります。 (2) サスペンド状態からリセットされる場合ホスト/Hubは最低でも10msのリセット信号を送出しますので、10ms以内に応答の準備を整える必要があります。 M66291はSuspendから復帰するのに差動信号の変化を検出します。割り込みは制御するCPUの状態によって変わりますがごく短い時間で発生します。 時間規定の詳細については、USB SpecificationのChapter 7を参照ください。
9.8.10.	[質問]	割り込み端子が、非アクティブからアクティブにアサートとされるとCPUが割り込み要因をクリアするまでそのアクティブのレベルが維持されているのでしょうか。
	[回答]	そのとおりです。
9.8.11.	[質問]	STALLの処理がよくわからないのですが。
	[回答]	STALL送出は基本的にホストからのリクエストエラー応答で使用します。また、SetFeature/ClearFeatureの両リクエストによるSTALL処理及び、デバイスクラス固有(ストレージクラスなど)の使用方法があります。 M66291では応答PID="STALL"設定を行うことでSTALL応答が可能です。
9.8.12.	[質問]	エンドポイント1~6の場合、応答PID="STALL"を設定した後、シーケンスビットをクリア(SQCLRの該当ビット='1'書き込み)して待ち、ホストからのClearFeatureリクエストで指定エンドポイントのSTALLを解除すれば良いのでしょうか。
	[回答]	エンドポイントi=1~6場合は、シーケンスビットクリア後ClearFeatureリクエストを待ちます。M66291では応答PID="BUF"/"NAK"に設定することでSTALL解除が可能です。
9.8.13.	[質問]	バルク、インタラプト等の違いによるSTALL解除法の違いはないのですか。
	[回答]	転送種別に伴う違いはありません。但し、デバイスクラスによってSTALL使用方法が異なりますのでご注意ください。M66291では応答PID="STALL"設定を行うことでSTALL応答が可能です。
9.8.14.	[質問]	M66291からの割り込み出力は、その割り込みを許可している要因が存在するかぎりレベル出力されるのでしょうか？
	[回答]	許可されている要因が回避されるまでレベル出力されます。
9.8.15.	[質問]	割り込み通知を禁止している要因でも、割り込みステータスフラグはセットされますか。
	[回答]	M66291は、割り込み通知を禁止している要因でも、割り込みステータスを設定します。
9.8.16.	[質問]	M66291のプロセスはCMOSですかNMOSですか。
	[回答]	CMOSです。従って入力端子に中間電位を与える、又は、Vbus、TEST、D+、D-以外の入力端子をオープンのままにすると、リーク電流が発生する可能性があります。システム設計の際にはご注意くださいようお願いいたします。
9.8.17.	[質問]	リモートウェイクアップ制御を行なう時、デバイスがレジューム状態から復帰(DVSQが変化)した事を簡単に確認する方法はありますか？
	[回答]	以下の設定を行う事で、レジューム状態から復帰した時にデバイスステート遷移割り込みが発生する事ができます。 DVSE ビットは常に"1"に設定しておき、サスペンド状態に移行した場合でも DVSE ビットはクリアしないようにする。 (リモートウェイクアップ制御を行なわない場合は、サスペンド状態に移行する際、DVSE ビットはクリアし、レジューム時に DVSE="1"をセットして下さい。)

改訂記録

M66291 アプリケーションノート

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2002.01.23	-	初版発行
1.10	2003.01.16	-	M66291HP の追加に伴う記述変更
		2	発振子の制限事項追加(テクニカルニュース ASSP-05-0205 の反映)
		37,39	FIFO アクセス改善
		42	サスペンド検出処理改善
		44	レジューム検出処理改善
		76,82, 86,87	F A Q更新(9.1.4, 9.1.7, 9.1.8, 9.5.7, 9.8.5, 9.8.17)
		-	M66291 レジスタ一覧追加
		-	誤記修正
1.11	2004.03.01	-	ルネサス様式に対応
		-	三菱表記をルネサス表記に変更
		49,50	SH ファミリの記述を追加
		49,53	H8 シリーズの記述を追加
		8,29,30, 31,39	誤記修正

M66291 レジスタ一覧

番地	レジスタ名	シンボル	奇数番地(001h)							偶数番地(000h)								
			bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0 0	USB動作許可レジスタ	USB_ENABLE	XCKE	PLL	Xtal[1:0]	SCKE	USBPC	Tr_on[1:0]							USBE			
0 2	リモートウェイクアップレジスタ	REMOTE_WAKEUP																
0 4	シーケンスビットクリア	SEQUENCE_BIT	SQCLR[6:0]															
0 6																		
0 8	USB Addressレジスタ	USB_ADDRESS	USB Addr[6:0]															
0 A	Isosynchronousステータスレジスタ	ISOSYNCHRONOUS_STATUS	OVN	CRCE	FMODE			FRNM[10:0]										
0 C	SOF制御レジスタ	SOF_CNT	SOFOE	SOFA														
0 E	極性設定レジスタ	POLARITY_CNT	VB01	RM01	SF01	DS01	CT01	BE01	NR01	RD01								
1 0	割り込み許可レジスタ0	INT_ENABLE0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	INTNE	INTRE	URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
1 2	割り込み許可レジスタ1	INT_ENABLE1	EPB_RE[6:0]															
1 4	割り込み許可レジスタ2	INT_ENABLE2	EPB_NRE[6:0]															
1 6	割り込み許可レジスタ3	INT_ENABLE3	EPB_EMPE[6:0]															
1 8	割り込みステータスレジスタ0	INT_STATUS0	VBUS	RESM	SOFR	DVST	CTRTR	BEMP	INTN	INTR	Vbus	DVSQ[2:0]	VALID	CTSQ[2:0]				
1 A	割り込みステータスレジスタ1	INT_STATUS1	EPB_RDY[6:0]															
1 C	割り込みステータスレジスタ2	INT_STATUS2	EPB_NRDY[6:0]															
1 E	割り込みステータスレジスタ3	INT_STATUS3	EPB_EMP_OVR[6:0]															
2 0	Requestレジスタ	REQUEST_TYPE	bRequest[7:0]							bmRequestType[7:0]								
2 2	Valueレジスタ	REQUEST_VALUE	wValue[15:0]															
2 4	Indexレジスタ	REQUEST_INDEX	wIndex[15:0]															
2 6	Lengthレジスタ	REQUEST_LENGTH	wLength[15:0]															
2 8	コントロール転送制御レジスタ	CONTROL_TRANSFER	CTRR	Ctr_Rd_Buf_Nmb[5:0]					CTRW	Ctr_Wr_Buf_Nmb[5:0]								
2 A	EP0パケットサイズレジスタ	EP0_PACKET_SIZE	EP0_MXPS[6:0]															
2 C	自動応答制御レジスタ	AUTO_RESPONSE_CONTROL	ASCN ASAD															
2 E																		
3 0	EP0 FIFO選択レジスタ	EP0_FIFO_SELECT	RCNT							Octl	BSWP							ISEL
3 2	EP0 FIFO制御レジスタ	EP0_FIFO_CONTROL	EP0_PID[1:0]	IVAL	BCLR	E0req	CCPL	ODLN[8:0]										
3 4	EP0 FIFOデータレジスタ	EP0_FIFO_DATA	EP0_FIFO[15:0]															
3 6	EP0連続送信データ長レジスタ	EP0_SEND_LEN	SDLN[8:0]															
3 8																		
3 A																		
3 C																		
3 E																		
4 0	CPU FIFO選択レジスタ	CPU_FIFO_SELECT	RCNT							Octl	BSWP							CPU_EP[3:0]
4 2	CPU FIFO制御レジスタ	CPU_FIFO_CONTROL	IDLY	IVAL	BCLR	Creq	CPU_DTLN[10:0]											
4 4	CPU FIFOデータレジスタ	CPU_FIFO_DATA	CPU_FIFO[15:0]															
4 6	SIE FIFOステータスレジスタ	SIE_FIFO_STATUS	TGL	SCLR	Sreq	SIE_DTLN[10:0]												
4 8	D0 FIFO選択レジスタ	D0_FIFO_SELECT	BUST	DFORM	RWND	ACKA	REQA	INTM	DMAEN	BSWP	Octl	DMA0_EP[5:0]						
4 A	D0 FIFO制御レジスタ	D0_FIFO_CONTROL	TRCLR	TREN	IVAL	BCLR	Dreq	DMA_DTLN[10:0]										
4 C	D0 FIFOデータレジスタ	D0_FIFO_DATA	D_FIFO[15:0]															
4 E	D0 トランザクション回数	D0_TRNCNT	TRNCNT[15:0]															
5 0	D1 FIFO選択レジスタ	D1_FIFO_SELECT	BUST	DFORM	RWND	ACKA	REQA	INTM	DMAEN	BSWP	Octl	DMA1_EP[5:0]						
5 2	D1 FIFO制御レジスタ	D1_FIFO_CONTROL	TRCLR	TREN	IVAL	BCLR	Dreq	DMA_DTLN[10:0]										
5 4	D1 FIFOデータレジスタ	D1_FIFO_DATA	DMA_FIFO[15:0]															
5 6	D1 トランザクション回数	D1_TRNCNT	TRNCNT[15:0]															
5 8	FIFOステータスレジスタ	FIFO_STATUS	EPB_STS[6:0]															
5 A	ポート方向レジスタ	PORT_CNTL	PDIR[6:0]															
5 C	ポートデータレジスタ	PORT_DATA	PDAT[6:0]															
5 E	駆動電流調整レジスタ	I.ADJ	LDRV															
6 0	EP1コンフィグレジスタ0	EP1_0CONF	EP1_TYP[1:0]	EP1_DIR	EP1_ITMD	EP1_Buf_siz[3:0]	EP1_DBLB	EP1_RWMD	EP1_Buf_Nmb[5:0]									
6 2	EP1コンフィグレジスタ1	EP1_1CONF	EP1_PID[1:0]	EP1_NULM	EP1_ACLR	EP1_Octl	EP1_MXPS[9:0]											
6 4	EP2コンフィグレジスタ0	EP2_0CONF	EP2_TYP[1:0]	EP2_DIR	EP2_ITMD	EP2_Buf_siz[3:0]	EP2_DBLB	EP2_RWMD	EP2_Buf_Nmb[5:0]									
6 6	EP2コンフィグレジスタ1	EP2_1CONF	EP2_PID[1:0]	EP2_NULM	EP2_ACLR	EP2_Octl	EP2_MXPS[9:0]											
6 8	EP3コンフィグレジスタ0	EP3_0CONF	EP3_TYP[1:0]	EP3_DIR	EP3_ITMD	EP3_Buf_siz[3:0]	EP3_DBLB	EP3_RWMD	EP3_Buf_Nmb[5:0]									
6 A	EP3コンフィグレジスタ1	EP3_1CONF	EP3_PID[1:0]	EP3_NULM	EP3_ACLR	EP3_Octl	EP3_MXPS[9:0]											
6 C	EP4コンフィグレジスタ0	EP4_0CONF	EP4_TYP[1:0]	EP4_DIR	EP4_ITMD	EP4_Buf_siz[3:0]	EP4_DBLB	EP4_RWMD	EP4_Buf_Nmb[5:0]									
6 E	EP4コンフィグレジスタ1	EP4_1CONF	EP4_PID[1:0]	EP4_NULM	EP4_ACLR	EP4_Octl	EP4_MXPS[9:0]											
7 0	EP5コンフィグレジスタ0	EP5_0CONF	EP5_TYP[1:0]	EP5_DIR	EP5_ITMD	EP5_Buf_siz[3:0]	EP5_DBLB	EP5_RWMD	EP5_Buf_Nmb[5:0]									
7 2	EP5コンフィグレジスタ1	EP5_1CONF	EP5_PID[1:0]	EP5_NULM	EP5_ACLR	EP5_Octl	EP5_MXPS[9:0]											
7 4	EP6コンフィグレジスタ0	EP6_0CONF	EP6_TYP[1:0]	EP6_DIR	EP6_ITMD	EP6_Buf_siz[3:0]	EP6_DBLB	EP6_RWMD	EP6_Buf_Nmb[5:0]									
7 6	EP6コンフィグレジスタ1	EP6_1CONF	EP6_PID[1:0]	EP6_NULM	EP6_ACLR	EP6_Octl	EP6_MXPS[9:0]											

- ※M66290Aからの追加レジスタ
- ※M66290Aからの削除レジスタ
- ※M66290Aからの機能変更レジスタ

M66290Aからの削除レジスタについて

1	SCTR(H'00番地)	機能削除
2	OVN(H'0A番地)	機能削除
3	CRCE(H'0A番地)	機能削除
4	DMAMD(H'62,H'66,H'6A,H'6E,H'72,H'76番地)	Dn_FIFO_SELECTレジスタに設定機能を追加し、当該レジスタ削除

M66290Aからの機能変更レジスタについて

1	WKUP(H'02番地)	M66291は、リモートウェイクアップ信号出力後、Suspendに入る前のデバイスステートに移ります。
2	VBUS(H'18番地)	M66291は、内部クロック停止時(SCKE=0)でもVbus変化の割り込みを通知します。
3	RESM(H'18番地)	M66291は、内部クロック停止時(SCKE=0)でもresume変化の割り込みを通知します。
4	Vbus(H'18番地)	M66291は、内部クロック停止時(SCKE=0)でもVbusの状態をモニターします。
5	EPB_NRE(H'14番地)	isochronous転送に設定したendpointに対応するビットは、割り込み許可しないでください。
6	EPB_NRDY(H'1C番地)	isochronous転送に設定したendpointに対応するビットは、参照しないでください。
7	ISEL(H'30番地)	M66291は、ISELビットを変更してもEP0_FIFOバッファカウンタの初期化を行いません。
8	CPU_FIFO選択レジスタ(H'40番地)	M66291は、デフォルト設定(RWND=0)時、CPU_EP変更時にCPU_FIFOバッファカウンタの初期化を行いません
9	D0_FIFO選択レジスタ(H'48番地)	M66291は、デフォルト設定(RWND=0)時、DMA0_EP変更時にDMA0_EPバッファカウンタの初期化を行いません
10	BUST(H'48,H'50番地)	対DMA0の動作設定フラグ追加により、名称変更しました

安全設計に関するお願い

- ・弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

- ・本資料は、お客様が用途に応じた適切なルネサス テクノロジー製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジーが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
- ・本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジーは責任を負いません。
- ・本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジーは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジー半導体製品のご購入に当たりましては、事前にルネサス テクノロジー、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジーホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
- ・本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジーはその責任を負いません。
- ・本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジーは、適用可否に対する責任を負いません。
- ・本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。
- ・本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジー、ルネサス販売または特約店へご照会ください。
- ・本資料の転載、複製については、文書によるルネサス テクノロジーの事前の承諾が必要です。
- ・本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジー、ルネサス販売または特約店までご照会ください。